Knowledge-Based Systems 226 (2021) 107123

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A novel kernel-free least squares twin support vector machine for fast and accurate multi-class classification

Zheming Gao [a], Shu-Cherng Fang [b], Xuerui Gao [c], Jian Luo [d,*], Negash Medhin [e]

[a] College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China
[b] Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, USA
[c] Department of Mathematics, Shanghai University, Shanghai 200444, China
[d] School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China
[e] Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

ARTICLE INFO

Article history:
Received 28 November 2020
Received in revised form 29 March 2021
Accepted 3 May 2021
Available online 5 May 2021

Keywords:
Multi-class classification
Least squares twin support vector machine
Double well potential
Kernel-free SVM
Imbalanced data

ABSTRACT

Multi-class classification is an important and challenging research topic with many real-life applications. The problem is much harder than the classical binary classification, especially when the given data set is imbalanced. Hidden nonlinear patterns in the data set can further complicate the task of multi-class classification. In this paper, we propose a kernel-free least squares twin support vector machine for multi-class classification. The proposed model employs a special fourth order polynomial surface, namely the double well potential surface, and adopts the "one-verses-all" classification strategy. An ℓ_2 regularization term is added to accommodate data sets with different levels of nonlinearity. We provide some theoretical analysis of the proposed model. Computational results using artificial data sets and public benchmarks clearly show the superior performance of the proposed model over other well-known multi-class classification methods, in particular for imbalanced data sets.

1. Introduction

Multi-class classification has many real-life applications such as disease diagnosis [1], company credit ratings [2] and machinery faults detection [3]. As a powerful binary classification technique, the support vector machine (SVM) [4] has been extended for multi-class classification with different multi-class classification strategies, such as the "one-versus-all" (OVA) strategy. As one of the commonly-used strategies, the idea of OVA is simple but powerful [5]. However, the SVM with OVA strategy may not well handle the multi-class imbalanced data sets, which often exist in real-life problems. Recently, another effective binary classification method — the twin SVM (TSVM) model [6] was proposed and has also been extended for multi-class classification. Since TSVM captures each class of data individually instead of directly finding a separation hyperplane between classes, it may be more effective for classifying imbalanced data sets. Furthermore, the idea of least squares SVM model [7] can be adopted to improve the computational efficiency of TSVM. Therefore, we are motivated to propose a state-of-the-art fast and accurate TSVM-based model for classifying multi-class imbalanced data sets.

Proposed in late 1990's, the soft-margin SVM model [4] separates the data points into two classes by generating a separation hyperplane, while maximizing the margin between the two classes and minimizing the misclassification errors. As an alternative SVM model for binary classification, TSVM is formulated as two convex quadratic programming (QP) problems [6]. It generates two nonparallel hyperplanes and its main idea is to let each hyperplane stay close to one class of data and keep away from the other class of data points as far as possible. The computational efficiency of TSVM is further improved by the least squares TSVM (LSTSVM) model proposed by Kumar et al. [8]. Combining the idea of TSVM and the least squares SVM model [7], the LSTSVM model generates the non-parallel separation hyperplanes by solving the systems of linear equations.

While TSVM and LSTSVM work effectively for the data set in which each class is linearly recognizable, they may fail for the data set with each class being distributed in nonlinear patterns. To strengthen the capability for nonlinear cases, the TSVM model equipped with kernels (KTSVM) [6] and the corresponding least squares KTSVM (LSKTSVM) [8] were proposed. Both KTSVM and LSKTSVM classify the data points after mapping them on a higher dimensional feature space. The KTSVM and the LSKTSVM models work well for some nonlinear cases and have achieved much

* Corresponding author.
E-mail addresses: tonygaobasketball@hotmail.com (Z. Gao), fang@ncsu.edu (S.-C. Fang), rigel1224@163.com (X. Gao), luojian546@hotmail.com (J. Luo), ngmedhin@ncsu.edu (N. Medhin).

https://doi.org/10.1016/j.knosys.2021.107123
0950-7051/

success in many applications, but there are some disadvantages. First, to the best of our knowledge, there is no general principle to pre-select a suitable kernel for a given data set. Moreover, the performance of these kernel-based TSVM model depends heavily on the parameters of the kernels.

Recently, a kernel-free quadratic TSVM (QTSVM) model and a corresponding least squares QTSVM (LSQTSVM) were proposed for binary classification in [9]. Without using kernels, two quadratic surfaces are directly generated, such that each surface is closer to one class of data points and keeps far away from the other class of data points. Although the QTSVM and LSQTSVM models work well for some cases, the quadratic surfaces they produce may not well handle the data set which is highly nonlinearly distributed. A recently-proposed special category of degree-4 polynomial function — double well potential (DWP) function [10,11] attracts our attention. As a fourth degree polynomial function, the DWP function has the form in which a quadratic term is embedded in a quadratic function. The DWP surface has been adopted in [12] for nonlinear binary classification. By taking advantage of its flexibility, we are motivated to build a kernel-free TSVM model that utilizes DWP surfaces to handle the data with high nonlinearity.

Binary classification methods can be extended to solve multi-class classification problems based on multi-class classification strategies. One of the most popular strategies for multi-class classification is the one-versus-all (OVA) strategy. For the OVA strategy, K binary classifiers are constructed to solve a K-class classification problem. And each binary classifier is obtained by using the binary classification method to distinguish one class of data points from the remaining classes of data points. There are some other commonly-used multi-class classification strategies proposed in literature, such as one-versus-one (OVO), all-versus-one (AVO) and so forth. Even though the OVA strategy is conceptually simple, it was proved to be effective [5] and relatively more efficient.

Another difficulty in multi-class classification is the issue of imbalanced data, which has been defined as a challenge in data mining [13]. Recall that, linear TSVM models have been extended for multi-class classification, but may not well handle the data sets with nonlinear patterns. Although kernels can be utilized for those nonlinear cases, they have shortages. The recently-proposed kernel-free QTSVM models overcome the drawbacks of the kernel-based TSVM models, however, they may not effectively capture highly nonlinear data patterns. In addition, the twin SVM model generates a surface for each class, it may capture an imbalanced data set better than other SVM models. Hence, in this paper, we are motivated to propose a novel ℓ_2 regularized least squares kernel-free TSVM model based on DWP surfaces and the OVA strategy for multi-class classification, which is denoted as reg-LSDWPTSVM. Computational experiments are conducted to investigate the effectiveness and efficiency of the proposed model along with other benchmark models. Besides, additional computational experiments are conducted to investigate the performance of the reg-LSDWPTSVM model on imbalanced data. The main contributions of this paper are summarized as follows.

1. For multi-class classification, the computational results indicate the superior performance of the proposed reg-LSDWPTSVM model over other benchmark models in terms of classification accuracy and efficiency. Especially, it has increasing dominance over other benchmark models as the data becomes more imbalanced.
2. The proposed reg-LSDWPTSVM model combines the idea of least squares kernel-free TSVM with quartic surfaces. It may save the efforts from selecting a proper kernel and tuning related kernel parameters, which avoids the shortages induced by kernel-based TSVM models. An ℓ_2 regularization term with the trade-off parameter is added to adjust the impact of the fourth order term of the DWP surfaces, which helps the proposed model to handle data sets with different levels of nonlinearity.
3. The proposed reg-LSDWPTSVM model is theoretically and numerically investigated in this paper. It is capable of capturing the embedded nonlinearity of a data set and it outperforms those kernel-based TSVM models and kernel-free TSVM models in literature. In addition, the computational results show the increasingly dominant performance of the proposed model in terms of accuracy as the number of data features increases. The computational efficiency of the proposed reg-LSDWPTSVM model is also verified.

The rest of the paper is organized as follows. We briefly introduce some notations and some related SVM models in Section 2. In Section 3, we introduce the DWP function and propose a least squares DWP-based TSVM model with ℓ_2 regularization. Some theoretical properties of the proposed model are also studied. The computational experiments are conducted to test the proposed model on some artificial and public benchmark data sets in Section 4. Section 5 concludes the paper.

2. Preliminary

In this section, we first introduce the notations that are used through out this paper in Section 2.1. Then a brief review of some related SVM and TSVM models is provided in Section 2.2.

2.1. Notations

Throughout this paper, scalars are denoted by lower case letters, vectors are denoted by bold lower case letters, and matrices are denoted by bold upper case letters. The n-dimensional real space is denoted by $\mathbb{R}^n$ and the n-dimensional nonnegative orthant is denoted by $\mathbb{R}^n_+$. The m by n zero matrix is denoted by $\mathbf{0}_{m\times n}$, the n-dimensional identity matrix is denoted by $\mathbf{I}_n$. The all-one matrix of size $m \times n$ is denoted by $\mathbf{1}_{m\times n}$, and the diagonal matrix with vector $\boldsymbol{c} = [c_1, \ldots, c_n]^T$ on its diagonal is denoted by $\mathrm{Diag}(c_1, \ldots, c_n)$. The set of all $n \times n$ real symmetric matrices is denoted by $\mathbb{S}^n$. $\forall \mathbf{B} \in \mathbb{S}^n$, we write $\mathbf{B} \succ 0$ if matrix $\mathbf{B}$ is positive definite, and $\mathbf{B} \succeq 0$ if matrix $\mathbf{B}$ is positive semidefinite. $\forall \mathbf{A} \in \mathbb{R}^{m\times n}$, we denote the ith row of $\mathbf{A}$ as $A_{i\bullet}$ and denote the jth column of $\mathbf{A}$ as $A_{\bullet j}$.

Let $r \triangleq \frac{n(n+1)}{2}$. For any symmetric matrix $\mathbf{A} \in \mathbb{S}^n$, all information of $\mathbf{A}$ is included in the upper triangular r elements [14–16]. Hence, the half-vectorization of the symmetric matrix $\mathbf{A}$ is defined in [16] as the following:

$$\mathrm{hvec}(\mathbf{A}) \triangleq [A_{11}, \ldots, A_{1n}, A_{22}, \ldots, A_{2n}, \ldots, A_{n-1,n-1}, A_{n-1,n}, A_{nn}]^T \in \mathbb{R}^r.$$

For any vector $\boldsymbol{a} = [a_1, \ldots, a_n]^T \in \mathbb{R}^n$, the vector of the cross terms between its elements is denoted by $\mathrm{lvec}(\boldsymbol{a})$ [14–16]:

$$\mathrm{lvec}(\boldsymbol{a}) \triangleq \left[\frac{1}{2}a_1a_1, a_1a_2, \ldots, a_1a_n, \frac{1}{2}a_2a_2, a_2a_3, \ldots, a_2a_n, \ldots, \frac{1}{2}a_na_n\right]^T \in \mathbb{R}^r.$$

Notice that $\frac{1}{2}\boldsymbol{a}^T\mathbf{A}\boldsymbol{a} = \mathrm{lvec}(\boldsymbol{a})^T\mathrm{hvec}(\mathbf{A})$, hence a quadratic term with respective to $\boldsymbol{a}$ can be substituted by a linear term.

For any supervised classification problem, the K-class data set can be mathematically denoted by

$$\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)_{i=1,\ldots,N} \mid \boldsymbol{x}^{(i)} \in \mathbb{R}^n,\ y^{(i)} \in \{1, \ldots, K\}\right\}, \quad (1)$$

where N is the amount of data points, n is the number of features, $\boldsymbol{x}^{(i)} \triangleq [x_1^{(i)}, \ldots, x_n^{(i)}]^T \in \mathbb{R}^n$ is the vector of n feature values of point i, and $y^{(i)}$ is the label of point $\boldsymbol{x}^{(i)}$. For class k, denote its index set as $I^k \triangleq \{k_1, \ldots, k_{N_k}\}$ where N_k is the number of data points in class k. Also, denote the data subset of class k as $\mathcal{D}^k$ such that

$$\mathcal{D}^k \triangleq \left\{ \boldsymbol{x}^{(i)} \in \mathbb{R}^n \middle| i \in I^k \right\}. \tag{2}$$

From definition, it is obvious that $\sum_{k=1}^K N_k = N$ and $\bigcup_{k=1}^K I^k = \{1, \ldots, N\}$. Let $\mathbf{X} \triangleq [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}]$. For further convenience, we define data matrices $\mathbf{X}_k$ and $\mathbf{X}_{\text{-}k}$.

$$\mathbf{X}_k \triangleq \left[\boldsymbol{x}^{(k_1)}, \ldots, \boldsymbol{x}^{(k_{N_k})} \right] \in \mathbb{R}^{n \times N_k}, \quad \forall k_p \in I^k, \quad p = 1, \ldots, N_k. \tag{3}$$

$$\mathbf{X}_{\text{-}k} \triangleq \left[\boldsymbol{x}^{(j_1)}, \ldots, \boldsymbol{x}^{(j_{N_{\text{-}k}})} \right] \in \mathbb{R}^{n \times (N_{\text{-}k})}, \quad \forall j_p \in I^{\text{-}k}, \quad p = 1, \ldots, N_{\text{-}k}. \tag{4}$$

where $N_{\text{-}k} \triangleq N - N_k$ and $I^{\text{-}k} \triangleq \{1, \ldots, N\} \setminus I^k$.

Given a data set of two classes and the predicted labels by a classifier, all data points can be partitioned into four groups: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Denoted as R, the true positive rate is defined by the following:

$$R \triangleq \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

2.2. Some related SVM models

In this subsection, we provide a brief introduction of SVM and TSVM models. The ordinary soft-margin SVM model is introduced in Section 2.2.1. We introduce the TSVM and LSTSVM models in Section 2.2.2. In Section 2.2.3, the QTSVM and LSQTSVM models are introduced.

2.2.1. Soft-margin SVM

The soft margin SVM (SSVM) model was originally proposed in [17] for binary classification. Given a binary data set $\mathcal{D}$ defined in (1) with $K = 2$, and let $\hat{y}^{(i)} \triangleq 2y^{(i)} - 1$ (i.e. $\hat{y}^{(i)} \in \{-1, 1\}$), the SSVM model is formulated as the following convex quadratic programming (QP) problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{u}\|_2^2 + C\sum_{i=1}^N \xi_i \\ s.t. \quad & \hat{y}^{(i)}\left(\boldsymbol{u}^T\boldsymbol{x}^{(i)} + d\right) \geqslant 1 - \xi_i, \quad \forall i = 1, \ldots, N \\ & \boldsymbol{u} \in \mathbb{R}^n, d \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N. \end{aligned} \tag{SSVM}$$

where $C > 0$ is the penalty parameter for data points. It produces a hyperplane $\mathcal{H} = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{u}^T\boldsymbol{x} + d = 0\}$ while maximizing the margin between two classes of points [4]. Also, the soft-margin idea [4] is adopted by introducing the slack vector $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^T \in \mathbb{R}^N$.

However, most of the data sets may be more appropriately separated by a nonlinear classifier instead of a linear one. It can be done by SSVM equipped with a kernel (KSVM) [4], which can be formulated as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{v}\|_2^2 + C\sum_{i=1}^N \xi_i \\ s.t. \quad & \hat{y}^{(i)}\left(\boldsymbol{v}^T\phi(\boldsymbol{x}^{(i)}) + d\right) \geqslant 1 - \xi_i, \quad \forall i = 1, \ldots, N \\ & \boldsymbol{v} \in \mathbb{R}^m, d \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N. \end{aligned} \tag{KSVM}$$

where ϕ maps data point $\boldsymbol{x}^{(i)}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$ ($n < m$) and $\mathcal{K}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \phi(\boldsymbol{x}^{(i)})^T\phi(\boldsymbol{x}^{(j)})$ is a kernel for any $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$. The most popular kernel used in literature is the radial basis function (RBF) kernel (also called Gaussian kernel). The idea of kernel-based SVM model is to first map the data points onto a higher dimensional feature space and then separate the mapped data points with a hyperplane in the higher dimensional feature space.

Notice that, (KSVM) is also a convex QP problem, it is important to study its dual problem, which can be formulated as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \mathcal{K}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\hat{y}^{(i)}\hat{y}^{(j)}\alpha_i\alpha_j - \sum_{i=1}^N \alpha_i \\ s.t. \quad & \sum_{i=1}^N \alpha_i\hat{y}^{(i)} = 0 \\ & 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, N. \end{aligned} \tag{DKSVM}$$

where C is the given parameter of penalties. The details of derivation for this dual problem can be found in [17]. Even though the dual gap is zero from the duality theory, the dual problem (DKSVM) is simpler as it has only one linear equality constraint with the upper bounds of variables. Hence, the (KSVM) model is usually trained from its dual side. One of the most popular approaches proposed in literature for solving its dual problem is the sequential minimal optimization (SMO) algorithm [18], which has been adopted in software packages such as LIBSVM [19].

2.2.2. TSVM and LSTSVM

The twin SVM (TSVM) model proposed in [6] for binary classification by using two non-parallel hyperplanes. Given a binary data set $\mathcal{D}$ as defined by (1) with $K = 2$, the binary classification task can be accomplished by solving two QP problems

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X}_1^T\boldsymbol{u}_1 + d_1\mathbf{1}_{N_1}\right\|_2^2 + C_1\mathbf{1}_{N_{\text{-}1}}^T\boldsymbol{\xi}^1 \\ s.t. \quad & -\left(\mathbf{X}_{\text{-}1}^T\boldsymbol{u}_1 + d_1\mathbf{1}_{N_{\text{-}1}}\right) \geqslant \mathbf{1}_{N_{\text{-}1}} - \boldsymbol{\xi}^1 \\ & \boldsymbol{u}_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}, \boldsymbol{\xi}^1 \in \mathbb{R}_+^{N_{\text{-}1}}. \end{aligned} \tag{TSVM^1}$$

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X}_2^T\boldsymbol{u}_2 + d_2\mathbf{1}_{N_2}\right\|_2^2 + C_2\mathbf{1}_{N_{\text{-}2}}^T\boldsymbol{\xi}^2 \\ s.t. \quad & -\left(\mathbf{X}_{\text{-}2}^T\boldsymbol{u}_2 + d_2\mathbf{1}_{N_{\text{-}2}}\right) \geqslant \mathbf{1}_{N_{\text{-}2}} - \boldsymbol{\xi}^2 \\ & \boldsymbol{u}_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}, \boldsymbol{\xi}^2 \in \mathbb{R}_+^{N_{\text{-}1}}. \end{aligned} \tag{TSVM^2}$$

where C_1, C_2 are given positive parameters. For $k = 1, 2$, the first term in the objective function of problem (TSVM^k) is the functional margin of class k and the inequality constraint depicts that all the data points in the other class stay at least one unit away from hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{u}_k^T\boldsymbol{x} + d_k = 0\}$. The slack variable $\boldsymbol{\xi}^k$ is added to measure the errors of some data points that are closer than this minimum distance of one. (TSVM^k) generates the hyperplane by minimizing the functional margin and a penalty term of the errors. Let $(\boldsymbol{u}_1^*, d_1^*, \boldsymbol{\xi}^{1^*})$ and $(\boldsymbol{u}_2^*, d_2^*, \boldsymbol{\xi}^{2^*})$ be optimal solution tuples to problems (TSVM^1) and (TSVM^2), respectively. Denote the hyperplane produced by model (TSVM^1) as $\mathcal{H}_1 \triangleq \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x}^T\boldsymbol{u}_1^* + d_1^* = 0\}$ and the hyperplane produced by model (TSVM^2) as $\mathcal{H}_2 \triangleq \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x}^T\boldsymbol{u}_2^* + d_2^* = 0\}$. Then a new data sample $\boldsymbol{x} \in \mathbb{R}^n$ is assigned to class k if the closest hyperplane it lies to is $\mathcal{H}_k$ for $k = 1, 2$. In general, the new data sample will be assigned to class $\hat{k}$ such that

$$\hat{k} = \underset{k=1,2}{\operatorname{argmin}} \left\{ \left| \boldsymbol{x}^T\boldsymbol{u}_k^* + d_k^* \right| \right\}. \tag{6}$$

TSVM can also be equipped with RBF kernel to capture the nonlinearity of the data. Define the kernel matrix $\mathcal{K}(\mathbf{A}, \mathbf{B})$ with its ij-th element $(\mathcal{K}(\mathbf{A}, \mathbf{B}))_{ij} \triangleq \mathcal{K}(A_{\bullet i}, B_{\bullet j})$, where $\mathcal{K}$ is the kernel. The kernel-based twin SVM (KTSVM) for binary classification is

formulated as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{K}(\mathbf{X}_1,\mathbf{X})\tilde{\boldsymbol{u}}_1 + d_1\mathbf{1}_{N_1}\|_2^2 + C_1\mathbf{1}_{N_{-1}}^T\boldsymbol{\xi}^1 \\ s.t. \quad & -\left(\mathcal{K}(\mathbf{X}_{-1},\mathbf{X})\tilde{\boldsymbol{u}}_1 + d_1\mathbf{1}_{N_{-1}}\right) \geqslant \mathbf{1}_{N_{-1}} - \boldsymbol{\xi}^1 \\ & \tilde{\boldsymbol{u}}_1 \in \mathbb{R}^N, d_1 \in \mathbb{R}, \boldsymbol{\xi}^1 \in \mathbb{R}_+^{N_{-1}}. \end{aligned} \tag{KTSVM^1}$$

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{K}(\mathbf{X}_2,\mathbf{X})\tilde{\boldsymbol{u}}_2 + d_2\mathbf{1}_{N_2}\|_2^2 + C_2\mathbf{1}_{N_{-2}}^T\boldsymbol{\xi}^2 \\ s.t. \quad & -\left(\mathcal{K}(\mathbf{X}_{-2},\mathbf{X})\tilde{\boldsymbol{u}}_2 + d_2\mathbf{1}_{N_{-2}}\right) \geqslant \mathbf{1}_{N_{-2}} - \boldsymbol{\xi}^2 \\ & \tilde{\boldsymbol{u}}_2 \in \mathbb{R}^N, d_2 \in \mathbb{R}, \boldsymbol{\xi}^2 \in \mathbb{R}_+^{N_{-2}}. \end{aligned} \tag{KTSVM^2}$$

where C_1, C_2 are given positive parameters. Denote the optimal solution tuples to problems (KTSVM^1) and (KTSVM^2) as $(\tilde{\boldsymbol{u}}_1^*, d_1^*, \boldsymbol{\xi}^{1*})$ and $(\tilde{\boldsymbol{u}}_2^*, d_2^*, \boldsymbol{\xi}^{2*})$, respectively. A new data point $\boldsymbol{x}$ will be assigned to class $\hat{k}$ if the kth kernel surface is closest to $\boldsymbol{x}$.

$$\hat{k} = \underset{k=1,2}{\operatorname{argmin}} \left\{ \left| \mathcal{K}(\boldsymbol{x},\mathbf{X})\tilde{\boldsymbol{u}}_k^* + d_k^* \right| \right\}. \tag{7}$$

The efficiency of the TSVM model has been enhanced by modifying the inequality constraints as the equality constraints, so that a least squares TSVM (LSTSVM) model is proposed in [8].

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X}_1^T\boldsymbol{u}_1 + d_1\mathbf{1}_{N_1}\right\|_2^2 + \frac{C_1}{2}\boldsymbol{\xi}^{1T}\boldsymbol{\xi}^1 \\ s.t. \quad & \mathbf{X}_{-1}^T\boldsymbol{u}_1 + d_1\mathbf{1}_{N_{-1}} = \mathbf{1}_{N_{-1}} - \boldsymbol{\xi}^1 \\ & \boldsymbol{u}_1 \in \mathbb{R}^n, d_1 \in \mathbb{R}, \boldsymbol{\xi}^1 \in \mathbb{R}^{N_{-1}}. \end{aligned} \tag{LSTSVM^1}$$

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X}_1^T\boldsymbol{u}_2 + d_2\mathbf{1}_{N_2}\right\|_2^2 + \frac{C_2}{2}\boldsymbol{\xi}^{2T}\boldsymbol{\xi}^2 \\ s.t. \quad & \mathbf{X}_{-2}^T\boldsymbol{u}_2 + d_2\mathbf{1}_{N_{-2}} = \mathbf{1}_{N_{-2}} - \boldsymbol{\xi}^2 \\ & \boldsymbol{u}_2 \in \mathbb{R}^n, d_2 \in \mathbb{R}, \boldsymbol{\xi}^2 \in \mathbb{R}^{N_{-2}}. \end{aligned} \tag{LSTSVM^2}$$

where C_1, C_2 are given positive parameters. Denote the optimal solution tuples to problems (LSTSVM^1) and (LSTSVM^2) as $(\boldsymbol{u}_1^*, d_1^*, \boldsymbol{\xi}^{1*})$ and $(\boldsymbol{u}_2^*, d_2^*, \boldsymbol{\xi}^{2*})$, respectively. For $k = 1, 2$, let $\boldsymbol{v}_k^* = \begin{bmatrix} \boldsymbol{u}_k^* \\ d_k^* \end{bmatrix}$, $\mathbf{A}_k = \begin{bmatrix} \mathbf{X}_k \\ \mathbf{1}_{N_k}^T \end{bmatrix}$, $\mathbf{A}_{-k} = \begin{bmatrix} \mathbf{X}_{-k} \\ \mathbf{1}_{N_{-k}}^T \end{bmatrix}$ and $\mathbf{G}_k = \mathbf{A}_k\mathbf{A}_k^T + C_k\mathbf{A}_{-k}\mathbf{A}_{-k}^T$, then the optimal solutions have analytical forms:

$$\begin{aligned} \boldsymbol{v}_k^* &= C_k\mathbf{G}_k^{-1}\mathbf{A}_{-k}\mathbf{1}_{N_{-k}} \\ \boldsymbol{\xi}^{k*} &= \left(\mathbf{I}_{N_{-k}} - C_k\mathbf{A}_{-k}^T\mathbf{G}_k^{-1}\mathbf{A}_{-k}\right)\mathbf{1}_{N_{-k}} \end{aligned} \tag{8}$$

where $k = 1, 2$. The optimal solution in (8) requires the inverse of matrix $\mathbf{G}_k$. Although $\mathbf{G}_k \succeq 0$, it may be singular or ill-conditioned. To avoid this situation, a small perturbation term $\epsilon\mathbf{I}$ is added to $\mathbf{G}_k$. The hyperplane $\mathcal{H}_k$ for kth class is determined by (8). The assignment of a new data point $\boldsymbol{x}$ is the same as defined in (6).

Similarly to KTSVM, the LSTSVM model can be equipped with a kernel as well. LSKTSVM can be formulated as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{K}(\mathbf{X}_1,\mathbf{X})\tilde{\boldsymbol{u}}_1 + d_1\mathbf{1}_{N_1}\|_2^2 + \frac{C_1}{2}\boldsymbol{\xi}^{1T}\boldsymbol{\xi}^1 \\ s.t. \quad & \mathcal{K}(\mathbf{X}_{-1},\mathbf{X})\tilde{\boldsymbol{u}}_1 + d_k\mathbf{1}_{N_{-1}} = \mathbf{1}_{N_{-1}} - \boldsymbol{\xi}^1 \\ & \tilde{\boldsymbol{u}}_1 \in \mathbb{R}^N, d_1 \in \mathbb{R}, \boldsymbol{\xi}^1 \in \mathbb{R}^{N_{-1}}. \end{aligned} \tag{LSKTSVM^1}$$

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{K}(\mathbf{X}_2,\mathbf{X})\tilde{\boldsymbol{u}}_2 + d_2\mathbf{1}_{N_2}\|_2^2 + \frac{C_2}{2}\boldsymbol{\xi}^{2T}\boldsymbol{\xi}^2 \\ s.t. \quad & \mathcal{K}(\mathbf{X}_{-2},\mathbf{X})\tilde{\boldsymbol{u}}_2 + d_k\mathbf{1}_{N_{-2}} = \mathbf{1}_{N_{-2}} - \boldsymbol{\xi}^2 \\ & \tilde{\boldsymbol{u}}_2 \in \mathbb{R}^N, d_2 \in \mathbb{R}, \boldsymbol{\xi}^2 \in \mathbb{R}^{N_{-2}}. \end{aligned} \tag{LSKTSVM^2}$$

where C_1, C_2 are given positive parameters. Denote the optimal solution tuples to problems (LSKTSVM^1) and (LSKTSVM^2) as $(\tilde{\boldsymbol{u}}_1^*, d_1^*, \boldsymbol{\xi}^{1*})$ and $(\tilde{\boldsymbol{u}}_2^*, d_2^*, \boldsymbol{\xi}^{2*})$, respectively. For $k = 1, 2$, let $\boldsymbol{v}_k^* = \begin{bmatrix} \tilde{\boldsymbol{u}}_k^* \\ d_k^* \end{bmatrix}$, $\mathbf{B}_k = \begin{bmatrix} \mathcal{K}(\mathbf{X}_k,\mathbf{X}) \\ \mathbf{1}_{N_k}^T \end{bmatrix}$, $\mathbf{B}_{-k} = \begin{bmatrix} \mathcal{K}(\mathbf{X}_{-k},\mathbf{X}) \\ \mathbf{1}_{N_{-k}}^T \end{bmatrix}$, then optimal solutions have analytical forms:

$$\begin{aligned} \boldsymbol{v}_k^* &= C_k\left(\mathbf{B}_k\mathbf{B}_k^T + C_k\mathbf{B}_{-k}\mathbf{B}_{-k}^T\right)^{-1}\mathbf{B}_{-k}\mathbf{1}_{N_{-k}} \\ \boldsymbol{\xi}^{k*} &= \left(\mathbf{I}_{N_{-k}} - C_k\mathbf{B}_{-k}^T\left(\mathbf{B}_k\mathbf{B}_k^T + C_k\mathbf{B}_{-k}\mathbf{B}_{-k}^T\right)^{-1}\mathbf{B}_{-k}\right)\mathbf{1}_{N_{-k}} \end{aligned} \tag{9}$$

Similarly, an small perturbation term $\epsilon\mathbf{I}$ is usually added to $\mathbf{B}_k\mathbf{B}_k^T + C_k\mathbf{B}_{-k}\mathbf{B}_{-k}^T$ to avoid the possible singularity. The assignment of a new data point $\boldsymbol{x}$ is based on Eq. (7).

2.2.3. QTSVM and LSQTSVM

Although the KTSVM and LSKTSVM work well for both linear and some nonlinear types of data, they may have some disadvantages [9]. First, there is no general principle to select a suitable kernel for a given data set. Besides, the performance of KTSVM and LSKTSVM models is highly related to the parameters in the kernel [20]. To overcome these drawbacks, Gao et al. [9] proposed a quadratic kernel-free TSVM (QTSVM) model and a quadratic kernel-free least squares TSVM (LSQTSVM) model for binary classification. In these models, quadratic surfaces are produced instead of hyperplanes for binary classification. For $k = 1, 2$, the QTSVM model for binary classification can be formulated as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i\in I^k}\left|\frac{1}{2}\boldsymbol{x}^{(i)T}\mathbf{W}_k\boldsymbol{x}^{(i)} + \boldsymbol{x}^{(i)T}\boldsymbol{b}_k + c_k\right|^2 + C_k\sum_{j=1}^{N_{-k}}\xi_j^k \\ s.t. \quad & -\left(\frac{1}{2}\boldsymbol{x}^{(k_j)T}\mathbf{W}_k\boldsymbol{x}^{(k_j)} + \boldsymbol{x}^{(k_j)T}\boldsymbol{b}_k + c_k\right) \geqslant 1 - \xi_j^k, \\ & \forall k_j \in I^{-k} \quad j = 1, \ldots, N_{-k} \\ & \mathbf{W}_k \in \mathbb{S}^n, \boldsymbol{b}_k \in \mathbb{R}^n, c_k \in \mathbb{R}, \boldsymbol{\xi}^k \in \mathbb{R}_+^{N_{-k}}. \end{aligned} \tag{QTSVM^k}$$

where $C_k > 0$ is a given parameter. Denote the optimal solution tuple to problem (QTSVM^k) as $(\mathbf{W}_k^*, \boldsymbol{b}_k^*, c_k^*)$, the quadratic surface produced by (QTSVM^k) for class k is $\mathcal{Q}_k \triangleq \{\boldsymbol{x} \in \mathbb{R}^n | \frac{1}{2}\boldsymbol{x}^T\mathbf{W}_k^*\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b}_k^* + c_k^* = 0\}$.

Define $\boldsymbol{w}_k \triangleq \text{hvec}(\mathbf{W}_k)$, $\boldsymbol{s}^{(i)} \triangleq \text{lvec}(\boldsymbol{x}^{(i)})$ $(\forall i = 1, \ldots N)$, matrices $\mathbf{S}_k$ and $\mathbf{S}_{-k}$ as the following:

$$\mathbf{S}_k \triangleq \left[\boldsymbol{s}^{(k_1)}, \ldots, \boldsymbol{s}^{(k_{N_k})}\right] \in \mathbb{R}^{\frac{n(n-1)}{2}\times N_k}, \quad \forall k_p \in I^k, \quad p = 1, \ldots, N_k. \tag{10}$$

$$\mathbf{S}_{-k} \triangleq \left[\boldsymbol{s}^{(j_1)}, \ldots, \boldsymbol{s}^{(j_{N_{-k}})}\right] \in \mathbb{R}^{\frac{n(n-1)}{2}\times N_{-k}}, \quad \forall j_p \in I^{-k}, \quad p = 1, \ldots, N_{-k}. \tag{11}$$

To make (QTSVM^k) simple to be solved, the matrix variable $\mathbf{W}_k$ can be vectorized and model (QTSVM^k) can be reformulated as the following [9,14]:

$$\begin{aligned} \min \quad & \frac{1}{2}\left(\boldsymbol{w}_k^T\mathbf{S}_k\mathbf{S}_k^T\boldsymbol{w}_k + \boldsymbol{b}_k^T\mathbf{X}_k\mathbf{X}_k^T\boldsymbol{b}_k + c_k^2\right) + C_k\mathbf{1}_{N_{-k}}{}^T\boldsymbol{\xi}^k \\ s.t. \quad & -\left(\mathbf{S}_{-k}^T\boldsymbol{w}_k + \mathbf{X}_{-k}^T\boldsymbol{b}_k + c_k\mathbf{1}_{N_{-k}}\right) \geqslant \mathbf{1}_{N_{-k}} - \boldsymbol{\xi}^k \\ & \boldsymbol{w}_k \in \mathbb{R}^{\frac{n(n+1)}{2}}, \boldsymbol{b}_k \in \mathbb{R}^n, c_k \in \mathbb{R}, \boldsymbol{\xi}^k \in \mathbb{R}_+^{N_{-k}}. \end{aligned} \tag{QTSVM$'^k$}$$

where $C_k > 0$ is a given parameter. Denote the optimal solution tuple to problem (QTSVM$'^k$) as $(\boldsymbol{w}_k^*, \boldsymbol{b}_k^*, c_k^*)$, and a new data point $\boldsymbol{x}$ will be assigned to class $\hat{k}$ such that

$$\hat{k} = \underset{k=1,2}{\operatorname{argmin}} \left\{ \left| \frac{1}{2}\boldsymbol{x}^T\mathbf{W}_k^*\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b}_k^* + c_k^* \right| \right\}. \tag{12}$$

where $\mathbf{W}_k^* = \text{hvec}^{-1}(\boldsymbol{w}_k^*)$.

Adopting the idea of LSTSVM, the LSQTSVM model was also proposed in [21] for binary classification as follows.

$$\begin{aligned}\min\quad & \frac{1}{2}\sum_{i\in I^k}\left|\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{W}_k\boldsymbol{x}^{(i)}+\boldsymbol{x}^{(i)^T}\boldsymbol{b}_k+c_k\right|^2+\frac{C_k}{2}\sum_{j=1}^{N_{-k}}\left(\xi_j^k\right)^2\\ s.t.\quad & \frac{1}{2}\boldsymbol{x}^{(k_j)^T}\mathbf{W}_k\boldsymbol{x}^{(k_j)}+\boldsymbol{x}^{(k_j)^T}\boldsymbol{b}_k+c_k=1-\xi_j^k,\\ & \forall k_j\in I^{-k}\quad j=1,\ldots,N_{-k}\\ & \mathbf{W}_k\in\mathbb{S}^n,\boldsymbol{b}_k\in\mathbb{R}^n,c_k\in\mathbb{R},\boldsymbol{\xi}^k\in\mathbb{R}^{N_{-k}}.\end{aligned}\tag{LSQTSVM^k}$$

where $k=1,2$. And it can be reformulated as the following QP problem:

$$\begin{aligned}\min\quad & \frac{1}{2}\left(\boldsymbol{w}_k^T\mathbf{S}_k\mathbf{S}_k^T\boldsymbol{w}_k+\boldsymbol{b}_k^T\mathbf{X}_k\mathbf{X}_k^T\boldsymbol{b}_k+c_k^2\right)+\frac{C_k}{2}\boldsymbol{\xi}^{k^T}\boldsymbol{\xi}^k\\ s.t.\quad & \mathbf{S}_{-k}^T\boldsymbol{w}_k+\mathbf{X}_{-k}^T\boldsymbol{b}_k+c_k\mathbf{1}_{N_{-k}}=\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k\\ & \boldsymbol{w}_k\in\mathbb{R}^{\frac{n(n+1)}{2}},\boldsymbol{b}_k\in\mathbb{R}^n,c_k\in\mathbb{R},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.\end{aligned}\tag{LSQTSVM$'^k$}$$

where $C_k>0$ is a given parameter. Denote the optimal solution to (LSQTSVM$'^k$) as $(\boldsymbol{w}_k^*,\boldsymbol{b}_k^*,c_k^*)$, the assignment of a new data point $\boldsymbol{x}$ is based on Eq. (12). Let $\boldsymbol{v}_k^*\triangleq\begin{bmatrix}\boldsymbol{w}_k^*\\ \boldsymbol{b}_k^*\\ c_k^*\end{bmatrix}$ and define matrices

$$\mathbf{D}_k\triangleq\begin{bmatrix}\mathbf{S}_k\\ \mathbf{X}_k\\ \mathbf{1}_{N_k}^T\end{bmatrix},\quad \mathbf{D}_{-k}\triangleq\begin{bmatrix}\mathbf{S}_{-k}\\ \mathbf{X}_{-k}\\ \mathbf{1}_{N_{-k}}^T\end{bmatrix},\quad \mathbf{L}_k\triangleq\mathbf{D}_k\mathbf{D}_k^T+C_k\mathbf{D}_{-k}\mathbf{D}_{-k}^T.\tag{13}$$

where S_k and S_{-k} are defined by (10) and (11), respectively. $(\boldsymbol{v}_k^*,\boldsymbol{\xi}^{k*})$ has the following analytical form:

$$\begin{aligned}\boldsymbol{v}_k^*&=C_k\mathbf{L}_k^{-1}\mathbf{D}_{-k}\mathbf{1}_{N_{-k}}\\ \boldsymbol{\xi}^{k*}&=\left(\mathbf{I}_{N_{-k}}-C_k\mathbf{D}_{-k}^T\mathbf{L}_k^{-1}\mathbf{D}_{-k}\right)\mathbf{1}_{N_{-k}}\end{aligned}\tag{14}$$

Similarly, a small perturbation term $\epsilon\mathbf{I}$ is usually added to $\mathbf{L}_k$ to avoid the possible singularity.

3. Twin SVM model with DWP surface

As we introduced before, both (QTSVM$'^k$) and (LSQTSVM$'^k$) produce a quadratic surface for kth class of data points. They work well for some nonlinear cases, but still cannot capture the high nonlinearity inside the given data set. In this section, we first introduce the DWP function in Section 3.1. In Section 3.2, we propose the kernel-free TSVM and LSTSVM models based on the DWP surfaces. Some theoretical analysis of the proposed models are provided in Section 3.2.

3.1. Double well potential function

The DWP function is a special type of fourth order polynomial functions defined as follows. It attracts considerable attention in the field of quantum mechanics, where the DWP function was used as the numerical approximation to the generalized Ginzburg–Landau functional [21].

Definition 3.1 (*DWP Function*)**.** Let P be a real-value function defined on $\mathbb{R}^n$ such that

$$P(\boldsymbol{x})=\frac{1}{2}\left(\frac{1}{2}\|\mathbf{B}\boldsymbol{x}-\boldsymbol{c}\|_2^2-d\right)^2+\frac{1}{2}\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}+\boldsymbol{b}^T\boldsymbol{x}+q.\tag{15}$$

where $\mathbf{B}\in\mathbb{R}^{m\times n}$, $\boldsymbol{c}\in\mathbb{R}^m$, $d\in\mathbb{R}$, $\mathbf{A}\in\mathbb{S}^n$, $\boldsymbol{b}\in\mathbb{R}^n$, $q\in\mathbb{R}$.

In addition to the high nonlinearity, the DWP function has the form of embedding a quadratic term in a quadratic function. Hence, it is more tractable than other 4th order polynomial functions. Motivated by its high nonlinearity and amenability, we utilize the DWP surfaces to capture the hidden nonlinearity inside the data and improve the classification accuracy.

Given a DWP function P, and any data point $(\boldsymbol{x}^{(i)},y^{(i)})\in\mathcal{D}$ denoted by (1) , define

$$\zeta^{(i)}\triangleq\frac{1}{2}\|\mathbf{B}\boldsymbol{x}^{(i)}-\boldsymbol{c}\|_2^2-d.\tag{16}$$

Define $\boldsymbol{s}^{(i)}\triangleq\mathrm{lvec}(\boldsymbol{x}^{(i)})$, $\boldsymbol{w}_B\triangleq\mathrm{hvec}(\mathbf{B}^T\mathbf{B})$, $\boldsymbol{w}_{Bc}\triangleq\boldsymbol{c}^T\mathbf{B}$ and $c_d\triangleq\frac{1}{2}\boldsymbol{c}^T\boldsymbol{c}-d$, then we have $\zeta^{(i)}=\boldsymbol{s}^{(i)^T}\boldsymbol{w}_B-\boldsymbol{x}^{(i)^T}\boldsymbol{w}_{Bc}+c_d$. Define $\boldsymbol{z}^{(i)}\triangleq[\boldsymbol{s}^{(i)^T},\boldsymbol{x}^{(i)^T},1]^T$ and $\boldsymbol{w}_\zeta\triangleq[\boldsymbol{w}_B^T,\boldsymbol{w}_{Bc}^T,c_d]^T$. Therefore,

$$P(\boldsymbol{x}^{(i)})=\tilde{P}\left(\begin{bmatrix}\boldsymbol{z}^{(i)}\\ \boldsymbol{x}^{(i)}\end{bmatrix}\right)\triangleq\frac{1}{2}\boldsymbol{z}^{(i)^T}\boldsymbol{w}_\zeta\boldsymbol{w}_\zeta{}^T\boldsymbol{z}^{(i)}+\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{A}\boldsymbol{x}^{(i)}+\boldsymbol{b}^T\boldsymbol{x}^{(i)}+q.\tag{17}$$

where function $\tilde{P}:\mathbb{R}^{\frac{n(n+1)}{2}+2n+1}\to\mathbb{R}$ has a quadratic term with respect to $\boldsymbol{z}^{(i)}$ on $\mathbb{R}^{\frac{n(n+1)}{2}+n+1}$ and another quadratic term with respect to $\boldsymbol{x}^{(i)}$ on $\mathbb{R}^n$. With the similar vectorization procedure, denote $l\triangleq\frac{n(n+1)}{2}+n+1$, $m_l\triangleq\frac{l(l+1)}{2}$ and

$$\begin{aligned}&\boldsymbol{w}_W\triangleq\mathrm{hvec}(\boldsymbol{w}_\zeta\boldsymbol{w}_\zeta^T)\in\mathbb{R}^{m_l},\\ &\boldsymbol{w}_A\triangleq\mathrm{hvec}(\mathbf{A})\in\mathbb{R}^{n(n+1)/2},\\ &\boldsymbol{\eta}^{(i)}\triangleq\mathrm{lvec}(\boldsymbol{z}^{(i)})\in\mathbb{R}^{m_l}.\end{aligned}\tag{18}$$

Consequently, $P(\boldsymbol{x}^{(i)})$ equals to a linear function P_l with respect to $\boldsymbol{\eta}^{(i)}$, $\boldsymbol{s}^{(i)}$ and $\boldsymbol{x}^{(i)}$ in $\mathbb{R}^{m_l+l-1}$, i.e.,

$$P(\boldsymbol{x}^{(i)})=P_l\left(\begin{bmatrix}\boldsymbol{\eta}^{(i)}\\ \boldsymbol{s}^{(i)}\\ \boldsymbol{x}^{(i)}\end{bmatrix}\right)\triangleq\boldsymbol{\eta}^{(i)^T}\boldsymbol{w}_W+\boldsymbol{s}^{(i)^T}\boldsymbol{w}_A+\boldsymbol{x}^{(i)^T}\boldsymbol{b}+q.\tag{19}$$

In other words, we have following result:

Lemma 3.1. *A DWP function in $\mathbb{R}^n$ is equivalent to a linear function in $\mathbb{R}^{m_l+l-1}$, where $m_l=\frac{l(l+1)}{2}$ and $l=\frac{n(n+1)}{2}+n+1$.*

Remark. In P_l, the coefficient $\boldsymbol{w}_W$ keeps the information of the embedded quadratic term and the coefficient $\boldsymbol{w}_A$ keeps the quadratic term of the original DWP function.

With the definition in (18), $\forall i\in\{1,\ldots,N\}$, let $\mathbf{H}\triangleq[\boldsymbol{\eta}^{(1)},\ldots,\boldsymbol{\eta}^{(N)}]$ and define

$$\mathbf{H}_k\triangleq\left[\boldsymbol{\eta}^{(k_1)},\ldots,\boldsymbol{\eta}^{(k_{N_k})}\right]\in\mathbb{R}^{n\times N_k},\quad\forall k_p\in I^k,\quad p=1,\ldots,N_k.\tag{20}$$

$$\mathbf{H}_{-k}\triangleq\left[\boldsymbol{\eta}^{(j_1)},\ldots,\boldsymbol{\eta}^{(j_{N_{-k}})}\right]\in\mathbb{R}^{n\times N_{-k}},\quad\forall j_p\in I^{-k},\quad p=1,\ldots,N_{-k}.\tag{21}$$

In addition, define matrix $\mathbf{M}_n$ as the following:

$$\mathbf{M}_n\triangleq\begin{bmatrix}\mathbf{I}_{m_l} & \mathbf{0}_{m_l\times l}\\ \mathbf{0}_{m_l\times l} & \mathbf{0}_{l\times l}\end{bmatrix}\in\mathbb{R}^{(m_l+l)\times(m_l+l)}\tag{22}$$

3.2. DWPTSVM & LSDWPTSVM with ℓ_2 regularization for multi-class classification

In this subsection, we first propose a kernel-free TSVM model by directly using DWP surfaces for multi-class classification with OVA strategy [22], which is denoted as DWPTSVM. Then we propose a least squares twin SVM model based on DWPTSVM.

The idea of DWPTSVM is to find DWP surfaces $\mathcal{S}_1,\ldots,\mathcal{S}_K$ such that the data points in class k stays as close to $\mathcal{S}_k$ as possible,

where $k = 1, \ldots, K$. Given a data set $\mathcal{D}$ as denoted in (1) with K classes of data, a multi-class classifier based on K DWP surfaces, can be obtained by solving K problems. For $k = 1, \ldots, K$, the kth problem (DWPTSVM^k) can be formulated as the following:

$$\begin{aligned}
\min \quad & \frac{1}{2}\sum_{i\in I^k}\left|\frac{1}{2}\left(\frac{1}{2}\|\mathbf{B}_k\boldsymbol{x}^{(i)}-\boldsymbol{c}_k\|_2^2-d_k\right)^2+\frac{1}{2}\boldsymbol{x}^{(i)^T}\mathbf{A}_k\boldsymbol{x}^{(i)}+\boldsymbol{b}_k^T\boldsymbol{x}^{(i)}+q_k\right|^2 \\
& +C_k\sum_{j=1}^{N_{-k}}\xi_j^k \\
s.t. \quad & -\left(\frac{1}{2}\left(\frac{1}{2}\|\mathbf{B}_k\boldsymbol{x}^{(k_j)}-\boldsymbol{c}_k\|_2^2-d_k\right)^2+\frac{1}{2}\boldsymbol{x}^{(k_j)^T}\mathbf{A}_k\boldsymbol{x}^{(k_j)}+\boldsymbol{b}_k^T\boldsymbol{x}^{(k_j)}+q_k\right) \\
& \geqslant 1-\xi_j^k, \\
& \forall k_j\in I^{-k}\quad j=1,\ldots,N_{-k} \\
& \mathbf{B}_k\in\mathbb{R}^{m\times n},\boldsymbol{c}_k\in\mathbb{R}^m,d_k\in\mathbb{R}, \\
& \mathbf{A}_k\in\mathbb{S}^n,\boldsymbol{b}_k\in\mathbb{R}^n,q_k\in\mathbb{R},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{DWPTSVM^k}$$

where $C_k > 0$ is a given parameter. Model (DWPTSVM^k) may not be easily solved since there are matrix variables in this model. Moreover, as a quartic polynomial surface, the DWP surface may overfit the real-life training data set. Recall that, the DWP function can be equivalently reformulated as a linear function by (19). In addition, an ℓ_2 regularization term $\frac{\delta_k}{2}\|\boldsymbol{w}_W\|_2^2$ is added to overcome the overfitting problem, where $\delta_k > 0$ $(k = 1, \ldots, K)$ is a given parameter. Hence, model (DWPTSVM^k) can be reformulated as the following model (reg-DWPTSVM$'^k$):

$$\begin{aligned}
\min \quad & \frac{1}{2}\left(\boldsymbol{w}_W^T\mathbf{H}_k\mathbf{H}_k^T\boldsymbol{w}_W+\boldsymbol{w}_A^T\mathbf{S}_k\mathbf{S}_k^T\boldsymbol{w}_A+\boldsymbol{b}^T\mathbf{X}_k\mathbf{X}_k^T\boldsymbol{b}+q^2N_k\right) \\
& +\frac{\delta_k}{2}\|\boldsymbol{w}_W\|_2^2+C_k\mathbf{1}_{N_{-k}}^T\boldsymbol{\xi}^k \\
s.t. \quad & -\left(\mathbf{H}_{-k}^T\boldsymbol{w}_W+\mathbf{S}_{-k}^T\boldsymbol{w}_A+\mathbf{X}_{-k}^T\boldsymbol{b}+q\mathbf{1}_{N_{-k}}\right)\geqslant\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k \\
& \boldsymbol{w}_W\in\mathbb{R}^{m_l},\boldsymbol{w}_A\in\mathbb{R}^{\frac{n(n+1)}{2}},\boldsymbol{b}\in\mathbb{R}^n,q\in\mathbb{R},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{reg-DWPTSVM$'^k$}$$

where C_k and δ_k are given positive parameters. Model (reg-DWPTSVM$'^k$) can be simplified as follows. Define matrices $\mathbf{E}_k$ and $\boldsymbol{\Lambda}_k$ as the following:

$$\mathbf{E}_k\triangleq\begin{bmatrix}\mathbf{H}_k\\ \mathbf{S}_k\\ \mathbf{X}_k\\ \mathbf{1}_{N_k}^T\end{bmatrix},\quad \mathbf{E}_{-k}\triangleq\begin{bmatrix}\mathbf{H}_{-k}\\ \mathbf{S}_{-k}\\ \mathbf{X}_{-k}\\ \mathbf{1}_{N_{-k}}^T\end{bmatrix} \tag{23}$$

and let $\boldsymbol{v}_k = [\boldsymbol{w}_W^T, \boldsymbol{w}_A^T, \boldsymbol{b}^T, q]^T$, then model (reg-DWPTSVM$'^k$) can be reformulated as the following form:

$$\begin{aligned}
\min \quad & \frac{1}{2}\boldsymbol{v}_k^T\left(\mathbf{E}_k\mathbf{E}_k^T+\delta_k\mathbf{M}_n\right)\boldsymbol{v}_k+C_k\mathbf{1}_{N_{-k}}^T\boldsymbol{\xi}^k \\
s.t. \quad & -\mathbf{E}_{-k}^T\boldsymbol{v}_k\geqslant\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k \\
& \boldsymbol{v}_k\in\mathbb{R}^{m_l+l},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{reg-DWPTSVM$''^k$}$$

where $\mathbf{M}_n$ is defined by Eq. (22), C_k and δ_k are given positive parameters. The existence of the optimal solution, denoted as $(\boldsymbol{v}_k^*, \boldsymbol{\xi}^{k*})$, is shown in the following theorem.

Theorem 3.2. *For any given data set $\mathcal{D}$ as defined in (1), $C_k > 0$ and $\delta_k > 0$, there exists an optimal solution to (reg-DWPTSVM$''^k$), which achieves a finite optimum.*

Proof. Since $\mathbf{E}_k\mathbf{E}_k^T \succeq 0$ and $\mathbf{M}_n \succ 0$, (reg-DWPTSVM$''^k$) is a convex QP problem. For any $\hat{\boldsymbol{v}}_k \in \mathbb{R}^{m_l+l}$, let $\hat{\xi}_j^k = \max\{0, 1 - \mathbf{E}_{\bullet j}^T\hat{\boldsymbol{v}}_k\}$ for $j = 1, \ldots, N_{-k}$. Then it is obvious that $(\hat{\boldsymbol{v}}_k, \hat{\boldsymbol{\xi}}^k = [\hat{\xi}_1^k, \ldots, \hat{\xi}_{N_{-k}}^k]^T)$ solves (reg-DWPTSVM$''^k$). Moreover, the objective function is bounded below by zero. Thus, the optimal solution to problem (reg-DWPTSVM$''^k$) exists and it achieves a finite optimum. □

A new data point $\boldsymbol{x}$ will be assigned to class $\hat{k}$ if

$$\hat{k}=\underset{k=1,\ldots,K}{\operatorname{argmin}}\left\{\left|[\boldsymbol{\eta}^T,\boldsymbol{s}^T,\boldsymbol{x}^T,1]\boldsymbol{v}_k^*\right|\right\}. \tag{24}$$

where $\boldsymbol{s} = \mathrm{lvec}(\boldsymbol{x})$ and $\boldsymbol{\eta} = \mathrm{lvec}(\boldsymbol{z})$ with $\boldsymbol{z} = [\boldsymbol{s}^T, \boldsymbol{x}^T, 1]^T$.

To improve the efficiency of (reg-DWPTSVM$''^k$), we propose the following kernel-free least squares TSVM with ℓ_2 regularization, by incorporating the least squares idea of LSTSVM.

$$\begin{aligned}
\min \quad & \frac{1}{2}\left(\boldsymbol{w}_W^T\mathbf{H}_k\mathbf{H}_k^T\boldsymbol{w}_W+\boldsymbol{w}_A^T\mathbf{S}_k\mathbf{S}_k^T\boldsymbol{w}_A+\boldsymbol{b}^T\mathbf{X}_k\mathbf{X}_k^T\boldsymbol{b}+q^2N_k\right) \\
& +\frac{\delta_k}{2}\|\boldsymbol{w}_W\|_2^2+\frac{C_k}{2}\boldsymbol{\xi}^{kT}\boldsymbol{\xi}^k \\
s.t. \quad & \mathbf{H}_{-k}^T\boldsymbol{w}_W+\mathbf{S}_{-k}^T\boldsymbol{w}_A+\mathbf{X}_{-k}^T\boldsymbol{b}+q\mathbf{1}_{N_{-k}}=\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k \\
& \boldsymbol{w}_W\in\mathbb{R}^{m_l},\boldsymbol{w}_A\in\mathbb{R}^{\frac{n(n+1)}{2}},\boldsymbol{b}\in\mathbb{R}^n,q\in\mathbb{R},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{reg-LSDWPTSVM$'^k$}$$

where C_k and δ_k are given positive parameters. Similarly, let $\boldsymbol{v}_k = [\boldsymbol{w}_W^T, \boldsymbol{w}_A^T, \boldsymbol{b}^T, q]^T$, problem (reg-LSDWPTSVM$'^k$) can be reformulated as the following:

$$\begin{aligned}
\min \quad & \frac{1}{2}\boldsymbol{v}_k^T\left(\mathbf{E}_k\mathbf{E}_k^T+\delta_k\mathbf{M}_n\right)\boldsymbol{v}_k+\frac{C_k}{2}\boldsymbol{\xi}^{kT}\boldsymbol{\xi}^k \\
s.t. \quad & \mathbf{E}_{-k}^T\boldsymbol{v}_k=\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k \\
& \boldsymbol{v}_k\in\mathbb{R}^{m_l+l},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{reg-LSDWPTSVM$''^k$}$$

where $\mathbf{M}_n$ is defined by Eq. (22). C_k and δ_k are given positive parameters. Denote the optimal solution as $(\boldsymbol{v}_k^*, \boldsymbol{\xi}^{k*})$, then the assignment of a new data point $\boldsymbol{x}$ is based on Eq. (24).

Define matrix $\mathbf{P}_k$ such that

$$\mathbf{P}_k\triangleq\mathbf{E}_k\mathbf{E}_k^T+\delta_k\mathbf{M}_n+\mathbf{E}_{-k}\mathbf{E}_{-k}^T. \tag{25}$$

Notice that, matrix $\mathbf{P}_k$ is positive semi-definite, so (reg-LSDWPTSVM$''^k$) is a convex QP problem.

Theorem 3.3. *Assume $\boldsymbol{P}_k \succ 0$, then there exists a unique optimal solution $(\boldsymbol{v}_k^*, \boldsymbol{\xi}^{k*})$ to problem (reg-LSDWPTSVM$''^k$) such that*

$$\begin{aligned}
\boldsymbol{v}_k^* &= C_k\boldsymbol{P}_k^{-1}\boldsymbol{E}_{-k}\mathbf{1}_{N_{-k}} \\
\boldsymbol{\xi}^{k*} &= \left(\boldsymbol{I}_{N_{-k}}-C_k\boldsymbol{E}_{-k}^T\boldsymbol{P}_k^{-1}\boldsymbol{E}_{-k}\right)\mathbf{1}_{N_{-k}}.
\end{aligned} \tag{26}$$

The proof of Theorem 3.3 is in Appendix.

However, the matrix $\mathbf{P}_k$ can be singular in some special cases. Even though the matrix yielded by the given data set $\mathcal{D}$ is positive definite in most cases, $\mathbf{P}_k$ can be ill-conditioned. A small perturbation term $\epsilon\mathbf{I}$ is usually added to avoid the possible singular or ill-conditioned matrix $\mathbf{P}_k$.

Theorem 3.4. *$\forall\epsilon > 0$, consider the following QP problem:*

$$\begin{aligned}
\min \quad & \frac{1}{2}\boldsymbol{v}_k^T\left(\boldsymbol{E}_k\boldsymbol{E}_k^T+\delta_k\boldsymbol{M}_n+\epsilon\boldsymbol{I}_{m_l+l}\right)\boldsymbol{v}_k+\frac{C_k}{2}\boldsymbol{\xi}^{kT}\boldsymbol{\xi}^k \\
s.t. \quad & \boldsymbol{E}_{-k}^T\boldsymbol{v}_k=\mathbf{1}_{N_{-k}}-\boldsymbol{\xi}^k \\
& \boldsymbol{v}_k\in\mathbb{R}^{m_l+l},\boldsymbol{\xi}^k\in\mathbb{R}_+^{N_{-k}}.
\end{aligned} \tag{reg-LSDWPTSVM$_\epsilon''^k$}$$

where $\boldsymbol{M}_n$ is defined by Eq. (22). C_k and δ_k are given positive parameters. Then the optimal solution exists and it is unique. Besides,

Table 1
Abbreviations and solvers of tested models.

Model	Abbreviation	Solver/Package	Parameters
Logistic regression	LR	Scikit-learn	–
Decision tree	DT	Scikit-learn	–
Kernel (RBF kernel) SVM	KSVM	LIBSVM	(C, γ)
Kernel (RBF kernel) twin SVM	KTSVM	Gurobi	(C_k, γ_k)
Least squares kernel (RBF kernel) twin SVM	LSKTSVM	–	(C_k, γ_k)
Quadratic surface twin SVM	QTSVM	Gurobi	C_k
Least squares quadratic surface twin SVM	LSQTSVM	–	C_k
Double well potential twin SVM with regularization	reg-DWPTSVM	Gurobi	(C_k, δ_k)
Least squares double well potential twin SVM with regularization	reg-LSDWPTSVM	–	(C_k, δ_k)

it has analytical form:

$$\begin{aligned} \boldsymbol{v}_k^* &= \frac{C_k}{\epsilon}\left[\boldsymbol{I}_{m_l+l} - \boldsymbol{Q}_k(\epsilon \boldsymbol{I}_{N_k} + \boldsymbol{Q}_k^T\boldsymbol{Q}_k)^{-1}\boldsymbol{Q}_k^T\right]\boldsymbol{E}_{\text{-}k}\mathbf{1}_{N_{\text{-}k}} \\ \boldsymbol{\xi}^{k*} &= \mathbf{1}_{N_{\text{-}k}} - \frac{C_k}{\epsilon}\boldsymbol{E}_{\text{-}k}^T\left[\boldsymbol{I}_{m_l+l} - \boldsymbol{Q}_k(\epsilon \boldsymbol{I}_{N_k} + \boldsymbol{Q}_k^T\boldsymbol{Q}_k)^{-1}\boldsymbol{Q}_k^T\right]\boldsymbol{E}_{\text{-}k}\mathbf{1}_{N_{\text{-}k}} \end{aligned} \tag{27}$$

where $\boldsymbol{Q}_k = [\boldsymbol{E}_k \ \sqrt{\delta_k}\boldsymbol{M}_n \ \boldsymbol{E}_{\text{-}k}]$.

The proof of 3.4 is in Appendix. Moreover, the flow of using reg-LSDWPTSVM for multi-class classification task is summarized as follows.

Model: reg-LSDWPTSVM

Training Phase
Input: data set $\mathcal{D}$ as defined in (2); parameters $C_i, \delta_i > 0$ $(i = 1, \ldots, K)$.
For $k = 1$ to K,
Calculate $\mathbf{X}_k$, $\mathbf{X}_{\text{-}k}$ as defined in (3) and (4).
Calculate $\mathbf{S}_k$, $\mathbf{S}_{\text{-}k}$ as defined in (10) and (11).
Calculate $\mathbf{E}_k$, $\mathbf{E}_{\text{-}k}$ and $\mathbf{P}_k$ as defined in (23) and (25).
Calculate and $\boldsymbol{v}_k^*$ and $\boldsymbol{\xi}^{k*}$ as defined in (26).
Output $\boldsymbol{v}_k^*$ and $\boldsymbol{\xi}^{k*}$ as the optimal solution of reg-LSDWPTSVM$'^k$.
Testing Phase
Assign the class label to a new data point by using the decision function in (24).

Remark. Notice that both the DWPSVM [12] and the proposed reg-LSDWPTSVM utilize the DWP surfaces, but there are several differences between them. First, they use different classification mechanisms. The DWPSVM generates a DWP surface as a separation surface, while the reg-LSDWPTSVM adopts the twin SVM [6] idea and generates a DWP surface to capture each class of data individually. The second difference is the goals. The DWPSVM was proposed for binary classification, while the reg-LSDWPTSVM is proposed to solve multi-class classification problems. Last but not the least, the DWPSVM does not have an analytical optimal solution, but the proposed reg-LSDWPTSVM model has the analytical optimal solution.

4. Computational experiments

In this section, we conduct computational experiments to investigate the performance of the proposed (reg-DWPTSVM$'^k$) and (reg-LSDWPTSVM$'^k$) models for multi-class classification. First, we introduce some settings of the experiments and show the flexibility of DWP surfaces in Section 4.1. Then we compare the twin SVM models and the least squares twin SVM models by conducting computational experiments on some artificial and public benchmarks in Section 4.2. More experiments are conducted to test the proposed reg-LSDWPTSVM model on some artificial data sets in Section 4.3 and some public benchmark data sets in Section 4.4.

Table 2
Data information.

Data set	Artificial data				Benchmark data	
	Arti2d3	Arti2d5	Arti3d3	Arti3d5	Iris	Seeds
(n, K)	(2, 3)	(2, 5)	(3, 3)	(3, 5)	(4, 3)	(7, 3)
# of data points	30 × 3	30 × 5	40 × 3	40 × 5	50 × 3	70 × 3

4.1. Experiment settings

In addition to the proposed reg-DWPTSVM model and reg-LSDWPTSVM model, some other twin SVM models are tested for comparisons, including TSVM model, LSTSVM model, KTSVM model, LSKTSVM model, QTSVM model and LSQTSVM model. To compare the twin SVM with traditional soft-margin SVM, the KSVM model is also tested. Moreover, we test the logistic regression (LR) model and the decision tree (DT) model along with others in the experiments since both LR and DT are widely used in the industry world for multi-class classification. All the SVM models are equipped with OVA strategy for multi-class classification in the experiments.

Throughout all tables and figures of results in this paper, each model is denoted by its abbreviation name, as listed in Table 1. All computational experiments are conducted on a desktop equipped with eight Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz CPUs and 8GB RAM. Moreover, we utilize Gurobi 8.1.1, LIBSVM [19] and Scikit-learn [23] to implement some of the tested models, as listed in Table 1.

For each tested data set, data points are normalized to [0, 1] to avoid the dominance of input features with greater numerical values over other smaller values. A 5-fold cross validation procedure is applied for each experiment and each experiment is repeated ten times for each model to make it statistically meaningful. All the possible parameters are tuned by using grid method, such as $\log_2 C \in \{-6, -3, \ldots, 21, 22\}$, $\log_2 \gamma \in \{-4, -3, \ldots, 3, 4\}$, $\log_2 C_k \in \{-8, -3, \ldots, 3, 4\}$ $(k = 1, \ldots, K)$, $\log_2 \gamma_k \in \{-8, -3, \ldots, 3, 4\}$ $(k = 1, \ldots, K)$ and $\log_2 \delta_k \in \{-8, -3, \ldots, 3, 4\}$ $(k = 1, \ldots, K)$.

In order to show and compare the flexibility of the DWP separation surfaces produced by the proposed reg-LSDWPTSVM model and other least squares twin SVM models, Figs. 1(a)–1(d) are displayed. The data set with linear, quadratic and highly nonlinear patterns is plotted and separated with different least square twin SVM models (LSTSVM, LSKTSVM, LSQTSVM and reg-LSDWPTSVM).

We have the following observations from the figures. In Fig. 1(a), LSTSVM model can only capture the linear pattern in the data set. In Fig. 1(b), LSQTSVM model captures both linear and quadratic patterns, but it does not capture the highly nonlinear pattern. In Figs. 1(c)–1(d), both reg-LSDWPTSVM and LSKTSVM models capture all the different patterns. And the reg-LSDWPTSVM model performs even better than the LSKTSVM model. In addition, the ℓ_2 regularization helps the reg-

Table 3
Artificial data results.

Model	Accuracy score %							
	Arti2d3		Arti2d5		Arti3d3		Arti3d5	
	mean/std	min/max	mean/std	min/max	mean/std	min/max	mean/std	min/max
LR	25.67/9.84	5.56/50.00	1.44/2.71	0.00/11.11	43.08/8.44	29.17/62.50	3.00/3.37	0.00/12.50
DT	54.78/10.77	27.78/77.78	38.22/11.75	11.11/66.67	64.17/7.19	50.00/79.17	39.50/7.95	20.83/58.33
KSVM	73.11/8.93	44.44/88.89	53.00/13.53	22.22/77.78	65.58/9.70	41.67/87.50	40.92/8.86	20.83/54.17
KTSVM	71.23/12.47	55.56/100.00	55.84/6.74	44.44/66.67	76.58/12.28	50.00/100.00	60.00/20.55	37.50/95.83
LSKTSVM	93.67/9.46	55.56/100.00	61.00/12.77	38.89/83.33	86.58/8.68	62.50/100.00	64.67/12.38	25.00/95.83
QTSVM	60.44/8.59	44.44/77.78	48.22/6.09	33.33/55.56	60.16/7.59	45.83/75.00	59.42/8.28	41.67/75.00
LSQTSVM	55.78/9.72	33.33/72.22	42.33/8.39	22.22/61.11	59.11/8.00	41.67/75.00	58.50/8.42	41.67/75.00
reg-DWPTSVM	99.56/1.52	94.44/100.00	98.33/5.17	72.22/100.00	94.58/4.23	83.33/100.00	91.67/5.32	83.33/100.00
reg-LSDWPTSVM	99.89/0.79	94.44/100.00	98.00/4.86	77.78/100.00	96.17/3.13	91.67/100.00	90.00/4.29	83.33/95.83

Table 4
Iris and seeds data results.

Model	Accuracy score %			
	Iris		Seeds	
	mean/std	min/max	mean/std	min/max
LR	96.67/3.37	86.67/100.00	94.29/3.49	85.71/100.00
DT	95.07/2.88	86.67/100.00	91.25/4.36	80.95/100.00
KSVM	95.60/3.12	86.67/100.00	94.46/3.07	85.71/97.62
KTSVM	95.53/3.47	86.67/100.00	89.76/4.25	80.95/97.62
LSKTSVM	96.87/2.81	90.00/100.00	95.60/3.03	85.71/100.00
QTSVM	96.53/3.36	90.00/100.00	92.62/4.34	83.33/100.00
LSQTSVM	96.73/2.38	93.33/100.00	94.52/3.29	85.71/100.00
reg-DWPTSVM	97.00/3.03	90.00/100.00	93.27/3.53	85.71/100.00
reg-LSDWPTSVM	97.47/2.57	90.00/100.00	96.25/2.95	90.48/100.00

LSDWPTSVM captures the quadratic patterns well without overfitting.

4.2. A comparison of twin SVM models and least squares twin SVM models

In this section, we would like to see the performances between those nonlinear twin SVMs and their least square versions.

All the models listed in Table 1 are tested on data sets listed in Table 2. For each data set, the accuracy scores and the average training CPU time of each tested model are recorded and some statistics are listed in Tables 3–5.

Here are some observations from the results.

- Comparing with all other tested models, the proposed reg-DWPTSVM and reg-LSDWPTSVM show their dominant performance on the artificial data sets in terms of classification accuracy. Since the artificial data sets are highly nonlinearly distributed, the results verify the flexibility of surfaces produced by the proposed reg-DWPTSVM and reg-LSDWPTSVM models.
- Among other tested benchmark models, the KTSVM and LSKTSVM perform relatively better. Indeed, they are able to capture some of the nonlinearity inside the data but cannot fit as well as the proposed reg-DWPTSVM and reg-LSDWPTSVM do. Although KSVM model can do nonlinear classification, it does not work effectively for the artificial data sets. The reason may be that these artificial data sets have high nonlinearity and they are more suitable for twin SVM models rather than soft-margin SVM models.
- Notice that, for some artificial data sets, the LSKTSVM model and the reg-LSDWPTSVM model produce higher mean accuracy scores than KTSVM and reg-DWPTSVM do, respectively. But the QTSVM model does perform better than LSQTSVM. Due to the produced nonlinear surfaces, the KTSVM and reg-DWPTSVM may cause overfitting, but their least squares models do not suffer the overfitting issue. For QTSVM, the quadratic surfaces it produces are not highly nonlinear, so that the overfitting may not easily happen. It explains the superior performance of QTSVM over LSQTSVM.
- For the iris data and the seeds data, they are relatively easy to be classified since the accuracy scores produced by all tested models are hardly less than 90%. Nevertheless, the proposed reg-LSDWPTSVM model still beat all other tested models. And the least squares version of each twin SVM model performs better than itself.
- The training CPU time of each twin SVM model is three magnitude orders bigger than that of its least squares version. Indeed, each twin SVM model is solved by Gurobi QP solver with interior point method. But each least squares SVM model has analytical solution and it can be obtained by solving a linear system, whose worst case computational complexity order is less than that of the interior point method. Moreover, the training CPU time consumed by the proposed reg-LSDWPTSVM is in the same order as those of other tested least squares twin SVM models.

4.3. Artificial data sets

In this section, more computational experiments are conducted on artificial data sets to see how the proposed reg-LSDWPTSVM model performs when the number of features increases and how it performs on imbalanced data sets. Since the analytical solutions provide much higher computational efficiency without losing much accuracy, only least squares twin SVM models along with LR, DT and KSVM models are tested as benchmarks in the computational experiments in the rest of this paper.

4.3.1. Artificial data with increasing number of features

We first test the proposed reg-LSDWPTSVM model on some artificial data sets of different numbers of features. Some basic information of the artificial data sets is listed in Table 6. For each class, we generate a nonlinear surface (one in quadratic form and two in quartic form with sufficient overlapping among them).

Table 5
Training CPU time.

Model	CPU time (s)					
	Arti2d3	Arti2d5	Arti3d3	Arti3d5	Iris	Seeds
LR	0.009	0.012	0.008	0.025	0.013	0.013
DT	0.003	< 0.001	0.002	0.006	< 0.001	< 0.001
KSVM	0.053	0.063	0.064	0.037	0.016	0.016
KTSVM	0.516	2.894	0.686	8.074	2.743	6.906
LSKTSVM	0.003	0.003	0.002	0.003	< 0.001	< 0.001
QTSVM	0.025	0.047	0.027	0.069	0.066	0.325
LSQTSVM	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
reg-DWPTSVM	0.103	0.166	0.599	0.96	3.532	185.229
reg-LSDWPTSVM	< 0.001	< 0.001	< 0.001	< 0.001	0.002	0.002

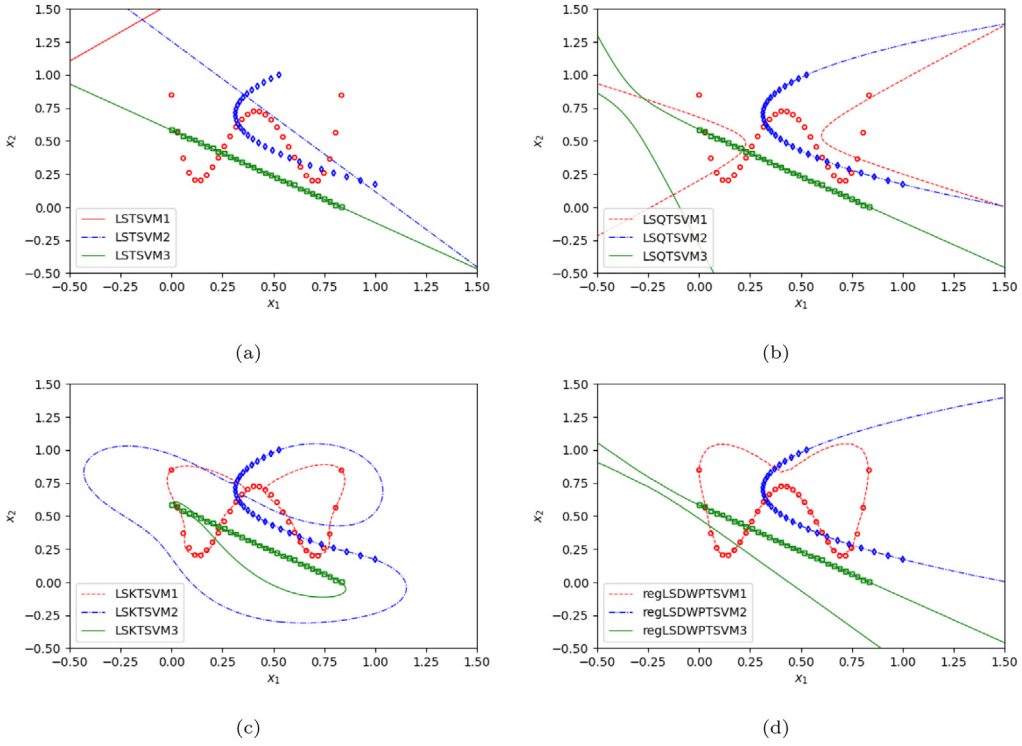

Fig. 1. Least squares twin SVM examples.

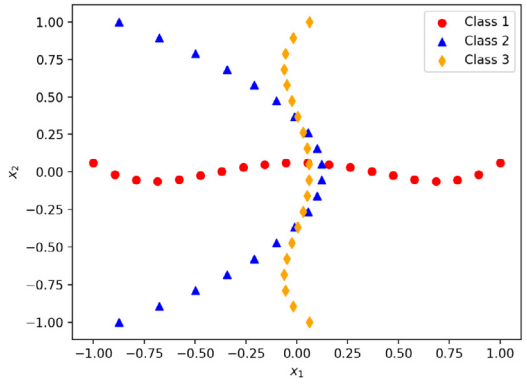

Fig. 2. B2d data set.

Then the data points are randomly selected on the generated nonlinear surface for each class. For example, the B2d data set is plotted in the following Fig. 2.

Table 6
Artificial B data information.

Data set	B2d	B4d	B6d	B8d	B10d
n	2	4	6	8	10
# of data points	20 × 3	100 × 3	240 × 3	450 × 3	880 × 3

The accuracy scores produced by all tested models are recorded and some statistics (mean, standard deviation, minimum and maximum) are listed in Tables 7–9. For each data set, we calculate the gap of the highest and the second highest mean accuracy scores (denoted as d%), and plot in Fig. 3.

From results listed in Tables 7–9, we observe that the proposed reg-LSDWPTSVM dominates all other tested models with highest mean accuracy scores and smallest standard deviations. More importantly, Fig. 3 shows the gap between the mean accuracy scores of reg-LSDWPTSVM and of the second most accurate model becomes bigger as the number of features increases. It also indicates that the proposed reg-LSDWPTSVM model may be more capable of capturing the nonlinearity of the data than LSKTSVM when the number of features increases.

Table 7
B2d and B4d results.

Model	B2d			B4d		
	Accuracy score %		CPU time (s)	Accuracy score %		CPU time (s)
	mean/std	min/max		mean/std	min/max	
LR	32.92/13.38	16.67/58.33	0.004	51.33/5.53	36.67/60.00	0.010
DT	82.08/13.59	58.33/100.00	0.001	79.22/4.75	71.67/85.00	0.003
KSVM	69.17/10.85	50.00/83.33	0.017	72.78/7.63	55.00/85.00	0.759
LSKTSVM	98.75/3.05	91.67/100.00	< 0.001	87.00/5.46	75.00/93.33	0.006
LSQTSVM	85.83/8.16	75.00/100.00	< 0.001	79.22/6.14	70.00/88.33	< 0.001
reg-LSDWPTSVM	100.00/0.00	100.00/100.00	< 0.001	92.33/2.94	86.67/96.67	0.004

Table 8
B6d and B8d results.

Model	B6d			B8d		
	Accuracy score %		CPU time (s)	Accuracy score %		CPU time (s)
	mean/std	min/max		mean/std	min/max	
LR	72.87/4.98	61.11/81.94	0.020	64.70/3.92	58.89/70.37	0.035
DT	80.60/3.68	74.31/86.81	0.003	85.30/4.51	78.15/92.59	0.006
KSVM	76.39/3.36	68.75/81.94	3.750	67.26/3.96	63.33/75.56	15.973
LSKTSVM	90.28/1.91	86.11/93.06	0.081	87.89/2.03	84.07/90.74	0.545
LSQTSVM	81.94/3.64	72.92/87.50	< 0.001	82.44/3.48	76.67/87.04	< 0.001
reg-LSDWPTSVM	97.13/1.36	94.44/99.31	0.035	97.28/0.58	96.39/98.06	0.180

Table 9
B10d results.

Model	B10d		
	Accuracy score %		CPU time (s)
	mean/std	min/max	
LR	77.40/1.59	73.67/80.11	0.044
DT	86.20/1.15	84.09/87.88	0.016
KSVM	85.58/1.73	83.14/87.88	50.068
LSKTSVM	86.16/1.72	83.90/89.96	1.610
LSQTSVM	82.56/1.15	80.87/84.47	0.001
reg-LSDWPTSVM	96.38/0.66	95.27/97.16	1.078

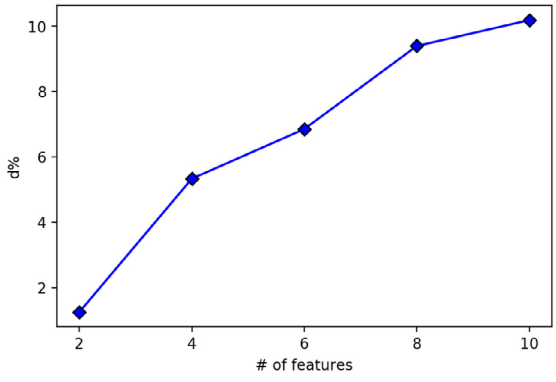

Fig. 3. d% vs. the number of features.

4.3.2. Artificial imbalanced data

In this subsection, we test the proposed reg-LSDWPTSVM model on some imbalanced artificial data sets with different number of features and imbalanced ratios. The artificial data sets utilized in the experiments are generated with 2, 4 and 8 features ($n = 2, 4, 8$) and each data set has 3 classes ($K = 3$). For each fixed n and the corresponding amount of data points (N), three data sets are created with imbalanced ratio to be 1, 3 and 10. More information of the artificial data sets is listed in Table 10.

Since the data sets are imbalanced, the accuracy score may not be a proper metric to distinguish the classification accuracy of different classes. Instead, the following average accuracy score (AvgAcc) defined by (28) is adopted to measure the classification accuracy of each model [24]:

$$\text{AvgAcc} \triangleq \frac{1}{K}\sum_{i=1}^{K} R_i \tag{28}$$

where R_i is the true positive rate (in percentage) of the ith class.

For each data set, the AvgAcc of each tested model is recorded and some statistics are listed in Tables 11–13.

Here are some observations from the results:

- The proposed reg-LSDWPTSVM model shows dominant performance over all other tested models in terms of average accuracy. In addition, with the same number of features, the mean AvgAcc produced by each tested model decreases when the imbalanced ratio increases. Indeed, more imbalanced the data is, more difficult it will be for classification.
- For a fixed number of features, the gap between the mean AvgAcc produced by reg-LSDWPTSVM and the second highest mean AvgAcc increases when the data set becomes more imbalanced. It indicates that even though the average accuracy scores could be affected when the imbalanced ratio increases, the proposed reg-LSDWPTSVM is more stable than others.

Table 10
Imbalanced artificial data information.

Data set	I1B2	I3B2	I10B2	I1B4	I3B4	I10B4	I1B8	I3B8	I10B8
n		2			4			8	
# of data points		180			420			1200	
$N_1/N_2/N_3$	60/60/60	108/36/36	150/15/15	140/140/140	252/84/84	350/35/35	400/400/400	720/240/240	1000/100/100
Imbalanced ratio	1	3	10	1	3	10	1	3	10

Table 11
Imbalanced data $n = 2$ results.

Model	I1B2		I3B2		I10B2	
	mean/std	min/max	mean/std	min/max	mean/std	min/max
LR	30.00/8.65	16.67/50.00	33.00/1.18	28.57/33.33	33.33/0.00	33.33/33.33
DT	67.30/7.95	55.56/86.11	58.24/9.85	34.52/78.57	46.18/11.05	31.11/76.67
KSVM	81.75/5.23	66.67/91.67	76.43/10.33	47.62/100.00	71.00/12.93	44.44/88.89
LSKTSVM	97.06/7.47	58.33/100.00	94.79/7.67	66.67/100.00	72.00/12.75	44.44/100.00
LSQTSVM	49.52/7.15	36.11/69.44	48.48/8.58	27.38/64.29	39.56/7.50	33.33/66.67
reg-LSDWPTSVM	100.00/0.00	100.00/100.00	99.71/1.14	95.24/100.00	79.37/9.61	66.67/100.00

Table 12
Imbalanced data $n = 4$ results.

Model	I1B2		I3B2		I10B2	
	mean/std	min/max	mean/std	min/max	mean/std	min/max
LR	79.40/4.55	67.86/91.67	50.07/4.10	42.58/55.75	48.57/5.41	42.86/57.14
DT	92.83/2.45	88.10/98.81	91.28/3.91	85.42/95.83	84.76/5.95	75.24/92.86
KSVM	89.12/3.45	83.33/100.00	85.63/6.58	73.75/96.58	76.76/7.92	64.76/95.24
LSKTSVM	96.62/2.95	85.71/100.00	93.24/4.56	85.42/97.92	78.14/9.83	61.90/90.00
LSQTSVM	92.26/2.80	86.90/100.00	91.29/4.84	79.92/97.92	53.24/6.13	42.86/66.19
reg-LSDWPTSVM	99.83/0.72	96.43/100.00	98.87/1.17	95.83/100.00	98.52/2.38	94.76/100.00

Table 13
Imbalanced data $n = 8$ results.

Model	I1B2		I3B2		I10B2	
	mean/std	min/max	mean/std	min/max	mean/std	min/max
LR	77.04/2.28	74.17/80.00	74.17/4.18	66.90/83.10	67.82/4.64	61.00/77.33
DT	89.83/2.24	85.83/93.75	88.87/4.87	81.25/94.21	80.83/5.85	72.67/88.83
KSVM	76.08/2.00	72.50/78.33	73.08/2.86	69.68/77.55	67.90/5.85	59.33/77.17
LSKTSVM	77.33/12.65	52.92/90.00	61.37/8.18	40.97/73.15	49.17/7.88	40.33/62.50
LSQTSVM	89.67/1.81	87.08/92.92	80.56/2.73	74.31/83.80	60.95/5.14	51.50/70.00
reg-LSDWPTSVM	94.03/1.45	90.83/96.25	93.65/1.41	92.13/96.30	91.85/3.56	87.00/98.83

4.4. Benchmark data sets

In this section, we investigate the performance of the proposed reg-LSDWPTSVM on some public benchmark data sets. The basic information of the benchmark data sets[1] are listed in Table 14.

In addition to the models listed in Table 1, one shallow artificial neural network (ANN) and one deep artificial neural network (DANN) are also tested for comparison. Besides the input and the output layers, the ANN has one hidden layer while the DANN has ten hidden layers. The number of nodes Ω on each hidden layer is decided by using the grid method $\Omega \in \{n, 2n, 4n\}$ in the training process. The activation functions in both ANN and DANN are rectified linear unit (ReLU) functions. Both ANN and DANN are implemented by using Keras 2.4.0 Python package.

Most of the public benchmark datasets are imbalanced so the AvgAcc is adopted to measure the classification accuracy. For each data set, the average training CPU time of each tested model and the AvgAcc scores are recorded and some statistics are listed in Tables 15–18. The CPU time consumed to forecast the class of each data point, which is denoted as the testing CPU time, is also recorded and listed in Table 18.

Here are some observations from the results:

- The proposed reg-LSDWPTSVM model produces higher mean AvgAcc scores than all other tested models on the tested benchmark data sets. Notice that, the second most accurate models on different data sets are different. For the Multiclass data set, it is the LR model; for the Breast tissue data, it is the DT model; for the Ecoli data set, it is LSQTSVM; for the Web phishing and Car evaluation data sets, it is the KSVM model; for the Wine data, it is the LSKTSVM model. However, they are all outperformed by the proposed reg-LSDWPTSVM model in terms of classification accuracy, which indicates that the reg-LSDWPTSVM model may work effectively on data sets of differently distributed patterns.
- Except the Multiclass and the Wine data sets, all the other data sets are imbalanced with the imbalanced ratio to be at least 3:1. Since the proposed reg-LSDWPTSVM model produces higher mean AvgAcc scores over all other tested models on all the tested data sets, we can see its effectiveness in classifying multi-class imbalanced data sets and its potential in solving real-life imbalanced multi-class classification problems.
- The training CPU time of the reg-LSDWPTSVM model on all data sets is acceptable. For all tested benchmark data sets, the proposed reg-LSDWPTSVM model is only 1–2 orders of magnitude slower than other tested models. Even though for

[1] Data sets are from UCI Machine Learning Repository [25] or kaggle.com.

Table 14
Benchmark data information.

Data set	Multiclass	Breast tissue	Wine	Ecoli	Web phishing	Car evaluation
n	8	9	13	7	9	6
K	3	3	3	5	3	4
# of data points	34/34/32	49/21/14	71/59/48	143/77/52/35/20	702/548/103	1210/384/69/65

Table 15
Multiclass & Breast tissue results.

Model	Multiclass			Breast tissue		
	AvgAcc %		CPU time (s)	AvgAcc %		CPU time (s)
	mean/std	min/max		mean/std	min/max	
LR	96.30/3.43	88.89/100.00	0.011	85.80/9.60	62.96/96.30	0.020
DT	91.48/4.63	83.33/100.00	0.001	88.95/9.00	67.59/100.00	0.003
KSVM	95.19/3.56	88.89/100.00	0.014	86.11/9.88	66.67/100.00	0.015
LSKTSVM	94.44/4.70	83.33/100.00	0.001	88.40/8.86	75.00/100.00	< 0.001
LSQTSVM	95.56/4.79	83.33/100.00	0.001	73.46/10.84	54.63/91.67	0.001
reg-LSDWPTSVM	97.41/3.56	88.89/100.00	0.159	89.69/8.76	75.00/100.00	0.548
ANN	94.67/7.08	66.67/100.00	0.386	89.04/7.67	75.93/100.00	0.709
DANN	95.00/5.52	83.33/100.00	23.995	89.17/10.29	67.59/100.00	22.911

Table 16
Wine & Ecoli results.

Model	Wine			Ecoli		
	AvgAcc %		CPU time (s)	AvgAcc %		CPU time (s)
	mean/std	min/max		mean/std	min/max	
LR	98.10/1.89	93.92/100.00	0.010	52.42/3.50	45.29/56.62	0.022
DT	91.93/5.62	80.42/100.00	0.002	49.10/3.53	43.19/56.43	0.002
KSVM	97.76/2.78	90.21/100.00	0.015	52.65/3.35	47.29/56.67	0.016
LSKTSVM	99.03/1.34	96.30/100.00	0.002	54.08/2.70	51.19/57.95	0.007
LSQTSVM	98.01/2.35	93.27/100.00	0.002	54.42/3.94	48.52/59.29	0.001
reg-LSDWPTSVM	99.16/1.36	95.24/100.00	6.481	54.60/2.54	49.95/58.00	0.132
ANN	97.43/2.88	89.56/100.00	0.184	51.27/3.57	45.19/57.33	2.795
DANN	98.04/3.32	89.56/100.00	42.531	51.53/5.49	44.43/58.00	69.752

Table 17
Web phishing & Car evaluation results.

Model	Web phishing			Car evaluation		
	AvgAcc %		CPU time (s)	AvgAcc %		CPU time (s)
	mean/std	min/max		mean/std	min/max	
LR	62.61/2.68	57.63/67.89	0.025	49.61/4.66	39.83/58.62	0.072
DT	85.75/3.50	79.43/90.89	< 0.001	70.82/2.32	65.35/74.67	< 0.001
KSVM	85.93/1.95	83.03/88.51	0.094	72.96/1.61	69.63/74.79	0.080
LSKTSVM	84.64/4.84	74.90/88.95	0.033	71.82/2.08	67.17/74.38	0.678
LSQTSVM	69.57/3.67	63.92/76.30	0.006	60.42/6.29	48.00/68.03	< 0.001
reg-LSDWPTSVM	86.79/3.35	82.90/92.28	0.423	73.23/1.12	71.18/74.38	0.067
ANN	83.36/5.55	66.60/91.30	7.664	69.80/4.96	52.51/74.57	5.812
DANN	83.65/4.47	74.36/89.39	247.555	72.84/1.10	71.76/74.67	263.783

Table 18
Testing CPU time on benchmark data.

Model	Testing CPU time (s)					
	Multiclass	Breast tissue	Wine	Ecoli	Web phishing	Car evaluation
LR	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
DT	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
KSVM	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
LSQTSVM	10^{-4}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
LSKTSVM	$< 10^{-4}$	10^{-3}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
reg-LSDWPTSVM	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
ANN	$< 10^{-4}$	10^{-4}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
DANN	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}

the Wine data, the CPU time of reg-LSDWPTSVM is larger than that for other data, it is still less than 7 s and acceptable. Besides, the mean AvgAcc score of reg-DWPTSVM is higher than that of the second most accurate model by 0.13%–1.11%. The advantage of reg-LSDWPTSVM may not be obvious, but it can be valuable in some real-life applications.

- Notice that, the Car evaluation data set has more than 1000 data points which is much more than the Ecoli data set. And its feature is only one less than that of the Ecoli data set. However, the training CPU time consumed by reg-LSDWPTSVM on Car evaluation data is less than that on Ecoli data. It indicates that the computational efficiency of

the proposed reg-LSDWPTSVM model is affected more by the number of data features than by the number of data points. The testing CPU time of the reg-LSDWPTSVM on each benchmark data set is short and acceptable.

4.5. Large-scale data sets

In this section, the proposed reg-LSDWPTSVM model is adjusted and modified in order to improve the computational efficiency when applied to high dimensional large-scale data sets. We first introduce how the proposed model is modified, and then conduct computational experiments to validate the adjusted model.

Recall that the convex QP problem (reg-LSDWPTSVM$''^k$) is solved for implementing the proposed reg-LSDWPTSVM model. However, from the computational results in Section 4.4, the training CPU time of the proposed model increases fast as the number of features of the data set increases. In some real-life applications, the data sets may have a large number of features, e.g., more than 100 features. In order to improve the computational efficiency when applied on those high dimensional large-scale data sets, the proposed model can be modified to reduce the computational complexity.

Similar to hvec and lvec defined in Section 2.1, we first define h^{d}vec and l^{d}vec as the following. For a symmetric matrix $\mathbf{A} \in \mathbb{S}^n$, and vector $\boldsymbol{a} \in \mathbb{R}^n$,

$$\mathrm{h}^d\mathrm{vec}(\mathbf{A}) \triangleq [A_{11}, A_{12}, A_{22}, A_{23}, \ldots, A_{n-1,n-1}, A_{n-1,n}, A_{nn}]^T \in \mathbb{R}^{2n-1}.$$

Notice that h^{d}vec($\mathbf{A}$) is different from hvec($\mathbf{A}$). Instead of keeping all the upper triangular information of the symmetric matrix $\mathbf{A}$, h^{d}vec($\mathbf{A}$) keeps only the main and the subdiagonal information of $\mathbf{A}$. It loses some information of $\mathbf{A}$, but reduces the dimension from $n(n-1)/2$ to be $2n-1$. In addition, define

$$\mathrm{l}^d\mathrm{vec}(\boldsymbol{a}) \triangleq \left[\frac{1}{2}a_1^2, a_1a_2, \frac{1}{2}a_2^2, a_2a_3, \ldots, \frac{1}{2}a_{n-1}^2, a_{n-1}a_n, \frac{1}{2}a_n^2\right]^T \in \mathbb{R}^{2n-1}.$$

Calculate the vectors in (10), (11) and (18) of the manuscript, by using the mappings h^{d}vec and l^{d}vec:

- $\hat{\boldsymbol{s}}^{(i)} \triangleq \mathrm{l}^d\mathrm{vec}(\boldsymbol{x}^{(i)}) \in \mathbb{R}^{2n-1}$, $\hat{\boldsymbol{z}}^{(i)} = [\hat{\boldsymbol{s}}^{(i)^T}, \boldsymbol{x}^{(i)^T}, 1]^T \in \mathbb{R}^{3n}$, $\hat{\boldsymbol{\eta}}^{(i)} = \mathrm{l}^d\mathrm{vec}(\hat{\boldsymbol{z}}^{(i)}) \in \mathbb{R}^{6n-1}$.
- $\hat{\boldsymbol{w}}_W = \mathrm{h}^d\mathrm{vec}(\boldsymbol{w}_\zeta \boldsymbol{w}_\zeta^T)$, $\hat{\boldsymbol{w}}_A = \mathrm{h}^d\mathrm{vec}(\mathbf{A})$.

And then calculate the corresponding data matrices $\hat{\mathbf{S}}_k$, $\hat{\mathbf{S}}_{\text{-}k}$, $\hat{\mathbf{H}}_k$, $\hat{\mathbf{H}}_{\text{-}k}$, $\hat{\mathbf{E}}_k$ and $\hat{\mathbf{E}}_{\text{-}k}$ as defined by (10), (11), (20), (21) and (23) in the manuscript with $\hat{\boldsymbol{s}}^{(i)}$, $\hat{\boldsymbol{\eta}}^{(i)}$. Moreover, calculate $\hat{\mathbf{M}}_n$ as the following:

$$\hat{\mathbf{M}}_n \triangleq \begin{bmatrix} \mathbf{I}_{6n-1} & \mathbf{0}_{(6n-1)\times 3n} \\ \mathbf{0}_{3n\times(6n-1)} & \mathbf{0}_{3n\times 3n} \end{bmatrix} \in \mathbb{R}^{(9n-1)\times(9n-1)} \tag{29}$$

Let $\hat{\boldsymbol{v}}_k = [\hat{\boldsymbol{w}}_W^T, \hat{\boldsymbol{w}}_A^T, \boldsymbol{b}^T, q]^T$, the proposed (reg-LSDWPTSVM$''^k$) is degenerated to be the following QP problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\hat{\boldsymbol{v}}_k^T\left(\hat{\mathbf{E}}_k\hat{\mathbf{E}}_k^T + \delta_k\hat{\mathbf{M}}_n\right)\hat{\boldsymbol{v}}_k + \frac{C_k}{2}\hat{\boldsymbol{\xi}}^{k^T}\hat{\boldsymbol{\xi}}^k \\ s.t. \quad & \hat{\mathbf{E}}_{\text{-}k}^T\hat{\boldsymbol{v}}_k = \mathbf{1}_{N_{\text{-}k}} - \hat{\boldsymbol{\xi}}^k \\ & \hat{\boldsymbol{v}}_k \in \mathbb{R}^{9n-1}, \hat{\boldsymbol{\xi}}^k \in \mathbb{R}_+^{N_{\text{-}k}}. \end{aligned} \qquad (\text{reg-LSDWP}^d\text{TSVM}''^k)$$

where C_k and δ_k are given positive parameters. Similarly, a new data point $\boldsymbol{x}$ will be assigned to class $\hat{k}$ if

$$\hat{k} = \mathrm{argmin}\{|\left[\hat{\boldsymbol{\eta}}^T, \hat{\boldsymbol{s}}^T, \boldsymbol{x}^T, 1\right]\hat{\boldsymbol{v}}_k^*|\}$$

Table 19
Large-scale data information.

Data set	GSAD	DNA	HARwS
(n, K)	(128, 5)	(180, 3)	(562, 5)
# of data points	197	1186	7352

Remark. (reg-LSDWP^{d}TSVM$''^k$) has only $9n - 1 + N_{\text{-}k}$ variables. Similarly, its optimal solution $\boldsymbol{v}_k^*$ can be obtained by solving a linear system with only $9n - 1$ variables and equations, whose worst case complexity is $\mathcal{O}(n^3)$. Hence, the computational cost will be much cheaper than that of the (reg-LSDWPTSVM$''^k$) model. Therefore, for high dimensional large-scale data sets, (reg-LSDWP^{d}TSVM$''^k$) is recommended to be implemented for (reg-LSDWPTSVM) for a better efficiency.

To validate the performance of the adjusted (reg-LSDWP^{d}TSVM$''^k$) model on large-scale data sets, we conduct computational experiments by using the following data sets (see Table 19)[2]:

The computational results are listed as the following. The results are also compared with other widely used multi-classifiers, including the LR, DT, KSVM and ANN.

From the results in Tables 20 and 21, we can see that the modified model provides not only the highest AvgAcc scores, but acceptable CPU time as well. This clearly indicates that the modified model has the potential in solving multi-class classification problems with high dimensional large-scale data sets.

5. Conclusion

In this paper, we have proposed an ℓ_2 regularized least squares kernel-free DWP twin SVM model with OVA strategy for multi-class classification. It directly produces the DWP surfaces. Certain theoretical properties have been studied. Computational experiments have been conducted to investigate the effectiveness and computational efficiency of the proposed model. Moreover, the proposed reg-LSDWPTSVM model has been tested on imbalanced data sets. Some major findings are summarized as follows.

- Equipped with the OVA strategy, the proposed reg-LSDWPTSVM model outperforms other least squares twin SVM models for multi-class classification problems in terms of classification accuracy. The computational results on imbalanced artificial data sets indicate an increasing dominance of the reg-LSDWPTSVM model over others when the data becomes more imbalanced. Moreover, the computational results on imbalanced public benchmarks imply the effectiveness of the proposed reg-LSDWPTSVM model in solving real-life imbalanced multi-class classification problems.
- The proposed reg-LSDWPTSVM model is a kernel-free SVM model, which does not require any kernel. It saves considerable effort when doing real-life applications. Moreover, it adopts the idea of the least squares SVM, which yields the satisfying computational efficiency. The DWP surfaces produced by reg-LSDWPTSVM are highly nonlinear quartic surfaces. Therefore, the proposed model has better capabilities to capture the hidden high-degree nonlinearity inside the data. In addition, the ℓ_2 regularization term on the fourth order term helps the produced surfaces fit the data with different levels of nonlinearity.

[2] Sources of data can be found in Appendix.

Table 20
Large-scale data results.

Model	AvgAcc %					
	GSAD		DNA		HARwS	
	mean/std	min/max	mean/std	min/max	mean/std	min/max
LR	58.18/2.74	50.00/60.00	92.44/1.25	90.68/94.92	72.77/0.51	71.92/73.39
DT	57.90/2.86	50.00/60.00	87.25/2.77	82.63/91.95	71.68/0.48	70.88/72.29
KSVM	58.20/2.61	51.00/60.00	93.39/1.91	89.41/96.61	72.03/0.53	71.05/72.78
reg-LSDWPTSVM	59.00/2.24	55.00/60.00	93.52/1.12	91.10/95.34	73.75/0.47	73.13/74.43
ANN	58.84/2.48	51.00/60.00	91.95/1.55	88.77/94.25	72.80/0.45	71.53/73.78

Table 21
Large-scale data CPU time.

Model	CPU time (s)		
	GSAD	DNA	HARwS
LR	0.015	0.018	0.691
DT	0.010	0.010	1.849
KSVM	0.037	0.355	6.687
reg-LSDWPTSVM	0.090	0.132	2.431
ANN	1.727	5.668	604.381

- The proposed reg-LSDWPTSVM model has shown its dominant performance on most of the artificial and public benchmark data sets. The computational results on some artificial data sets indicate its increasing dominance over other tested models as the number of features increases. Moreover, the computational results on public benchmark data sets show the strong potential of the reg-LSDWPTSVM model in solving real-life multi-class classification problems.
- The proposed reg-LSDWPTSVM has been adjusted and implemented for multi-class classification tasks with high dimensional data sets. Due to a simpler structure, the computational efficiency is significantly improved. Computational results on large-scale data sets have validated the promising performance of the modified model.

Our research can be extended to some additional research works. First, although the optimal solution to the proposed reg-LSDWPTSVM model has an analytical form, the training CPU time consumed increases fast as the size of the data set increases. So an immediate future work is to design an suitable algorithm [26,27] to improve the computational efficiency of the proposed model. Moreover, the proposed reg-LSDWPTSVM model can be extended for real-life applications with imbalanced data, including disease diagnosis [28] and credit scoring [29,30]. Also, we would like to investigate the robustness [31] of the proposed model.

CRediT authorship contribution statement

Zheming Gao: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Shu-Cherng Fang:** Supervision, Conceptualization, Writing - review & editing, Resources. **Xuerui Gao:** Writing - review & editing, Investigation. **Jian Luo:** Conceptualization, Methodology, Writing - original draft. **Negash Medhin:** Writing - review & editing, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been sponsored by the National Natural Science Foundation of China Grant #71701035.

Appendix. Proofs and the source of data

The following information can be found in this link:

https://github.com/tonygaobasketball/reg-LSDWPTSVM-project.

- Proofs of Theorems 3.3 and 3.4.
- Resources of public benchmark data sets used in Sections 4.4 and 4.5.

References

[1] H. Wang, B. Zheng, S.W. Yoon, H.S. Ko, A support vector machine-based ensemble algorithm for breast cancer diagnosis, European J. Oper. Res. 267 (2018) 687–699.
[2] K.-j. Kim, H. Ahn, A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach, Comput. Oper. Res. 39 (2012) 1800–1811.
[3] X. Zhang, B. Wang, X. Chen, Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine, Knowl.-Based Syst. 89 (2015) 56–85.
[4] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[5] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[6] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 905–910.
[7] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.
[8] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (2009) 7535–7543.
[9] Q.-Q. Gao, Y.-Q. Bai, Y.-R. Zhan, Quadratic kernel-free least square twin support vector machine for binary classification problems, J. Oper. Res. Soc. China 7 (2019) 539–559.
[10] S.-C. Fang, D. Gao, G.-X. Lin, R.-L. Sheu, W. Xing, Double well potential function and its optimization in the *n*-dimensional real space – part I, J. Ind. Manag. Optim. 13 (2017) 1291–1305, http://dx.doi.org/10.3934/jimo.2016073.
[11] Y. Xia, R.-L. Sheu, S.-C. Fang, W. Xing, Double well potential function and its optimization in the *n*-dimensional real space – part II, J. Ind. Manag. Optim. 13 (2017) 1307–1328, http://dx.doi.org/10.3934/jimo.2016074.
[12] Z. Gao, S.-C. Fang, J. Luo, N. Medhin, A kernel-free double well potential support vector machine with applications, European J. Oper. Res. 290 (2021) 248–262.
[13] Q. Yang, X. Wu, 10 challenging problems in data mining research, Int. J. Inf. Technol. Decis. Mak. 5 (2006) 597–604.
[14] J. Luo, S.-C. Fang, Z. Deng, X. Guo, Soft quadratic surface support vector machine for binary classification, Asia-Pac. J. Oper. Res. 33 (2016) 1650046.
[15] I. Dagher, Quadratic kernel-free non-linear support vector machine, J. Global Optim. 41 (2008) 15–30.
[16] S. Mousavi, Z. Gao, L. Han, A. Lim, Quadratic surface support vector machine with ℓ_1 norm regularization, 2019, arXiv preprint arXiv:1908.08616.
[17] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[18] J. Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Technical Report MSR-TR-98-14, 1998.
[19] C.-C. Chang, C.-J. Lin, Libsvm: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (2011) 1–27.
[20] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond, MIT Press, Cambridge, MA, USA, 2001.
[21] D.Y. Gao, H. Yu, Multi-scale modelling and canonical dual finite element method in phase transitions of solids, Int. J. Solids Struct. 45 (2008) 3660–3673.

[22] D. Tomar, S. Agarwal, A comparison on multi-class classification methods based on least squares twin support vector machine, Knowl.-Based Syst. 81 (2015) 131–147.
[23] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.
[24] A. Fernández, V. López, M. Galar, M.J. Del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, Knowl.-Based Syst. 42 (2013) 97–110.
[25] D. Dua, C. Graff, UCI machine learning repository, 2017, URL: http://archive.ics.uci.edu/ml.
[26] H. Cheng, P.-N. Tan, R. Jin, Efficient algorithm for localized support vector machine, IEEE Trans. Knowl. Data Eng. 22 (2009) 537–549.
[27] C.A. de Araúj Padilha, D.A.C. Barone, A.D.D. Neto, A multi-level approach using genetic algorithms in an ensemble of least squares support vector machines, Knowl.-Based Syst. 106 (2016) 85–95.
[28] X. Wang, Y. Yang, Y. Xu, Q. Chen, H. Wang, H. Gao, Predicting hypoglycemic drugs of type 2 diabetes based on weighted rank support vector machine, Knowl.-Based Syst. (2020) 105868.
[29] J. Luo, X. Yan, Y. Tian, Unsupervised quadratic surface support vector machine with application to credit risk assessment, European J. Oper. Res. 280 (2020) 1008–1017.
[30] S. Maldonado, J. Pérez, C. Bravo, Cost-based feature selection for support vector machines: An application in credit scoring, European J. Oper. Res. 261 (2017) 656–665.
[31] J. Ma, L. Yang, Q. Sun, Adaptive robust learning framework for twin support vector machine classification, Knowl.-Based Syst. 211 (2021) 106536.